IP Office

TAPI Link Programmierhandbuch

Inhaltsverzeichnis

IP Office TAPI Link 5
Überblick 5
IP Office TAPI-Treiber 5
Haftungsausschluss 5
Referenzmaterial 5
Struktur dieses Dokuments 6
Änderungen 6
Installieren der TAPILink- und Wave-Treiber 7
Installieren der CTI TAPILink Pro- und Wave-Lizenzen 7
Konfigurieren des TAPI-Treibers 7
Einzelbenutzermodus 7
Drittanwendermodus 7
ACD-Warteschlangen 7
Wave-Benutzer 8
Konfigurieren von IP Office für TAPI 8
Überblick 9
TAPI 2 9
TAPI 3 9

TAPI 2.x-Referenz 11
TAPI-Funktionen 11
lineAddToConference 12
lineAnswer 12
lineBlindTransfer 12
lineCompleteTransfer 12
lineConfigDialog 13
lineDeallocateCall 13
lineDevSpecific 14
lineDial 17
lineDrop 17
lineGenerateDigits 17
lineGenerateTone 17
lineGetAddressCaps 18
lineGetAddressID 18
lineGetAddressStatus 18
lineGetAppPriority 19
lineGetCallInfo 19
lineGetCallStatus 19
lineGetDevCaps 19
lineGetID 20
lineGetLineDevStatus 21
lineHandoff 24
lineHold 24
lineInitializeEx 24
lineMakeCall 24
lineMonitorDigits 25
lineMonitorTone 25
lineNegotiateAPIVersion 25
lineOpen 26
linePark 26
lineRedirect 26
lineRemoveFromConference 27
lineSetAppPriority 27
lineSetAppSpecific 27
lineSetCallPrivilege 27
lineSetStatusMessages 28

lineSetupTransfer ... 28
lineShutdown ... 28
lineSwapHold ... 28
lineUnhold ... 29
lineUnpark ... 29
TAPI-Strukturen ... 30
LINEADDRESSCAPS ... 30
LINEADDRESSSTATUS ... 36
LINECALLINFO ... 37
LINECALLPARAMS ... 38
LINECALLSTATUS ... 39
LINEDEVCAPS ... 40
TAPI-Ereignisse (Meldungen) ... 43
LINE_APPNEWCALL ... 43
LINE_CALLINFO ... 43
LINE_CALLSTATE ... 43
LINE_LINEDEVSTATE ... 43
LINE_DEVSPECIFIC ... 43
LINE_ADDRESSSTATE ... 43

TAPI 3.0 ... 45
TAPI ... 45
ITTAPI ... 45
Address ... 46
ITAddress ... 46
IEnumAddress ... 47
ITMediaSupport ... 47
Terminal ... 47
Call ... 48
ITCallInfo ... 48
ITBasicCallControl ... 49
ITCallStateEvent ... 52
ITCallNotificationEvent ... 53
ITCallInfoChangeEvent ... 53
CallHub ... 53

TAPI 3-Aufzählungstypen ... 55
CALL_STATE ... 55
CALLINFO_STRING ... 56
DISCONNECT_CODE ... 57
CALL_STATE_EVENT_CAUSE ... 57

IP Office Media Service Provider ... 59
Info zum MSP ... 59
Verwenden des MSP ... 59
Verwenden der gerätespezifischen Schnittstellen ... 59
ITACDAgent ... 60
ITGroup ... 60
ITDivert ... 61
ITPlay ... 62
IPOfficePrivateEvents ... 62
Verwenden der Media-Streaming-Funktionen des MSP ... 63

Index ... 65

IP Office TAPI Link

Überblick

IP Office CTI Link ist verfügbar in Lite- und Pro-Versionen, die den Anwendungen zur Laufzeit Schnittstellen zur Verfügung stellen. Das Software Development Kit (SDK) beinhaltet eine Dokumentation der Lite- und Pro-Schnittstellen für die Softwareentwickler.

Lite und Pro umfassen das gleiche Programm. Die zusätzlichen Funktionen von IP Office CTI Link Pro werden aktiviert, wenn der CTI Link Pro-Lizenzschlüssel installiert ist. Weitere Informationen finden Sie im IP Office CTI Link-Installationshandbuch.

Dieses Dokument enthält Informationen, die für Softwareentwickler bei der Implementierung einer Anwendung hilfreich sind, die den IP Office TAPI-Dienstanbieter verwendet. Es wird vorausgesetzt, dass der Entwickler bereits mit TAPI vertraut ist. Des Weiteren ist beim Lesen dieses Dokuments der Zugriff auf die Microsoft Developer Network-Bibliothek (MSDN) von Nutzen, die eine vollständige TAPI-Dokumentation beinhaltet.

IP Office TAPI-Treiber

Die Architektur von Windows ermöglicht Entwicklern, Anwendungen unter Verwendung von standardmäßigen Anwendungsprogrammierschnittstellen (Application Programming Interface, API) zu implementieren, unabhängig von den verwendeten Telefonieanlagen. Hersteller von Telefonieanlagen stellen Telefonietreiber (Telephony Service Provider, TSP) bereit, die unter Windows installiert werden. Diese TSPs liefern die Verbindung zwischen TAPI und den Telefonieanlagen.

Der TAPI-Treiber für IP Office unterstützt alle TAPI-Versionen von 2.0 bis 3.0.

Haftungsausschluss

Avaya ist darum bemüht, dass die Releases des IP Office TAPI-Treibers mit älteren Versionen des Treibers rückwärts kompatibel sind, kann jedoch nicht gewährleisten, dass der Betrieb von IP Office mit den bereitgestellten Funktionen genau gleich abläuft. Aufgrund von Verbesserungen in IP Office ist es wahrscheinlich, dass sich die genaue Reihenfolge, die zeitlichen Abläufe und die Inhalte der TAPI-Ereignisse ändern. Wir empfehlen Entwicklern, ein ereignisgesteuertes Programmiermodell zu verwenden, damit durch solche Änderungen keine Nachteile entstehen.

Referenzmaterial

Wir empfehlen folgendes Referenzmaterial:

- MSDN/Plattform SDK
- Windows Telephony Programming (TAPI 1.x und 2.x).
- CTI Link-Installationshandbuch

Struktur dieses Dokuments

Dieses Dokument besteht aus folgenden Abschnitten:

- TAPI 2
    - TAPI-Funktionen (in alphabetischer Reihenfolge)
    - TAPI-Strukturen (in alphabetischer Reihenfolge)
    - TAPI-Ereignisse (in alphabetischer Reihenfolge)
- TAPI 3
- TAPI 3-Aufzählungstypen
- Media Service Provider
- Verlust und Wiederherstellung der Kommunikation

Änderungen

Seit TAPI-Release 1.0.0.15 auf der Benutzer-CD 1.4 von IP Office wurden folgende Elemente hinzugefügt:

- Neue DevSpecific-Funktionen für TAPI 2. Die Funktion **lineDevSpecific** kann zur Bereitstellung von Funktionen zum Abhören/Aufschalten verwendet werden.
- Erweiterte Erläuterung der Funktion **lineGetLineDevStatus**.

Installieren der TAPILink- und Wave-Treiber

Der IP Office TAPI-Dienstanbieter und der Wave-Treiber werden mit Hilfe der Benutzer-CD von IP Office installiert.

Weitere Informationen finden Sie im CTI Link-Installationshandbuch.

Installieren der CTI TAPILink Pro- und Wave-Lizenzen

Sie benötigen keine Lizenz, um den TAPI-Treiber zu verwenden, die Lizenz bietet jedoch folgende zusätzliche Funktionen:

- Drittanwendermodus
- Überwachung der ACD-Warteschlange
- aktivierte Funktion **lineDevSpecific**

Zur Verwendung der Wave-Funktionalität müssen Sie zusätzlich zur CTI Link Pro-Lizenz eine *Wave-Benutzerlizenz* für jeden Wave-Benutzer installieren.

Konfigurieren des TAPI-Treibers

Der IP Office TAPI-Dienstanbieter kann im Einzelbenutzer- oder im Drittanwendermodus arbeiten. Für den Drittanwendermodus muss eine Lizenz erworben werden. Beachten Sie, dass Sie diese Option auch bei der nicht-lizenzierten Version (Lite) wählen können. Ohne Lizenz funktioniert sie jedoch nicht.

Einzelbenutzermodus bedeutet, dass mit der TAPI-Anwendung ein einzelnes Telefoniegerät gesteuert und/oder überwacht werden kann. Drittanwendermodus bedeutet, dass mit der TAPI-Anwendung alle Telefoniegeräte in einer bestimmten IP Office-Einheit gesteuert und/oder überwacht werden können.

Einzelbenutzermodus

Geben Sie die IP-Adresse der IP Office-Einheit in das Feld **IP-Adresse der Vermittlung** ein. Wählen Sie die Option **Einzelbenutzer**. Geben Sie den Benutzernamen und das Kennwort für die Nebenstelle ein, die von TAPI überwacht und/oder gesteuert werden soll. Der Benutzername ist normalerweise der Name einer Person, die einer physischen Nebenstelle zugeordnet ist.

Drittanwendermodus

Geben Sie die IP-Adresse der IP Office-Einheit in das Feld **IP-Adresse der Vermittlung** ein. Wählen Sie die Option **Drittanwender**. Geben Sie das Kennwort für die Vermittlung ein. Hierbei handelt es sich um das gleiche Kennwort, das beim Laden der Vermittlungs-Konfiguration im Manager eingegeben wird.

Im Drittanwendermodus wird standardmäßig eine TAPI-Leitung für jeden physischen IP Office-Anschluss bereitgestellt. Über die Kontrollkästchen für den Drittanwendermodus können zusätzliche Einheiten von TAPI überwacht und/oder gesteuert werden.

ACD-Warteschlangen

IP Office kann so konfiguriert werden, dass eingehende Anrufe, die an eine Gruppe interner Benutzer vermittelt werden, in eine Warteschlange gestellt werden. Wenn Ihr IP Office z.B. mit einer Gruppe von Call Center-Mitarbeitern konfiguriert wurde, können Sie einen eingehenden Anruf so lange in die Warteschlange stellen, bis ein Mitarbeiter den Anruf entgegennehmen kann.

Durch das Aktivieren des Kontrollkästchens **ACD-Warteschlangen** werden Leitungen zur Überwachung und/oder Steuerung der wartenden Anrufe für eine Gruppe eingerichtet.

Wave-Benutzer

Benutzer, deren Benutzername mit "TAPI:" beginnt, sind Wave-Benutzer. Die IP Office-Vermittlung versucht, Audio per Streaming an Wave-Benutzer weiterzuleiten, wenn diese an Anrufen beteiligt sind.

Für dieses Audio-Streaming muss der IP Office-Wave-Treiber auf dem PC installiert sein. Der Treiber wird für jeden Benutzer einzeln lizenziert. Falls der Wave-Treiber nicht installiert ist, haben Sie eventuell noch das Kontrollkästchen **Wave-Benutzer** aktiviert und erhalten weiterhin Wave-Benutzerereignisse, ohne dass eine Lizenz erforderlich ist.

Für den TAPI-WAV-Audio-Stream wird ein IP Office-Datenkanal aus dem gleichen Pool wie für VoiceMail-Ports verwendet. Die maximale Anzahl verfügbarer Datenkanäle für gleichzeitige VoiceMail- und TAPI-WAV-Anrufe hängt vom Typ der IP Office-Steuereinheit ab. IP401 = 2, IP403 und Small Office Edition = 10, IP406 = 20, IP412 = 30.

Konfigurieren von IP Office für TAPI

In diesem Abschnitt wird die Konfiguration von IP Office mittels der Manager-Anwendung beschrieben. Es gibt zwei Möglichkeiten, TAPI mit IP Office zu verwenden:

1. Wenn Ihre Anwendung Telefone überwacht, aber nicht steuert, ist keine Konfiguration notwendig.
2. Wenn Ihre Anwendung Telefone steuert, sollten Sie alle Benutzer, die gesteuert werden, als **Freisprechstation** konfigurieren. Dadurch wird das Telefon des Benutzers in den Ruhezustand versetzt, wenn ein Anruf mittels TAPI aufgelegt wird. Ist diese Option nicht aktiviert, bleibt das Telefon im nicht-verbundenen Zustand, bis es manuell aufgelegt wird. Das Kontrollkästchen **Freisprechstation** befindet sich auf der Registerkarte **Telefonie** der Benutzereinstellungen im Manager.
3. Wenn Sie möchten, dass Medienströme von einem bestimmten Benutzer gehandhabt werden (z.B. eine automatische Weiterleitung), erstellen Sie einen neuen Benutzer, dessen Name mit "TAPI:" beginnt. Damit kennzeichnen Sie ihn als Wave-Benutzer. Die Nummer dieses Benutzers sollte in einem Bereich liegen, der nicht mit vorhandenen Telefonnummern oder Gruppen in Konflikt tritt.

Überblick

Sie sollten alle TAPI-Anwendungen schließen, bevor Sie den Switch zurücksetzen. So kann der Telephony Service Provider (TSP) alle offenen Leitungen ordnungsgemäß schließen und es wird sichergestellt, dass der Switch und alle verbundenen TSPs einen einheitlichen Status aufweisen. Im Fall eines unerwarteten Kommunikationsverlusts (der Switch wird versehentlich heruntergefahren oder ein Netzwerkkabel wird versehentlich getrennt) erkennt der TSP, dass keine Verbindung mehr zum Switch besteht.

In diesem Zeitraum werden alle Aufrufe von TAPI-Funktionen abgelehnt, für die die Kommunikation zwischen TSP und Switch erforderlich ist. Die Zeit, die nach dem Kommunikationsverlust verstreicht, bis der TSP den Verlust erkennt, hängt von den TCP-Einstellungen auf dem Host und den TSP-internen Zeiteinstellungen ab. Es kann eine Verzögerung von bis zu zwei Minuten auftreten.

Nachdem der TSP erkannt hat, dass die Kommunikation mit dem Switch unterbrochen wurde, versucht er, die Verbindung wiederherzustellen. Bei erfolgreicher neuer Verbindung kann der Dienstanbieter die offenen Leitungen und Adressen in der Regel wiederherstellen. Dies ist auch dann der Fall, wenn der Kommunikationsverlust auf einen Neustart des Switch zurückzuführen ist.

Wie sich der Kommunikationsverlust für die TAPI-Anwendung darstellt, hängt von der verwendeten TAPI-Version ab, wie im Folgenden beschrieben.

TAPI 2

Wenn die Verbindung zwischen TSP und IP Office unterbrochen wird, wird über alle offenen Leitungen die Meldung LINEDEVSTATE_OUTOFSERVICE gesendet. Bei erfolgreicher Wiederherstellung der Verbindung wird für jede wiederhergestellte TAPI-Leitung die Meldung LINEDEVSTATE_INSERVICE gesendet.

TAPI 3

Wenn die Verbindung zwischen TSP und IP Office unterbrochen wird, wird für jede Adresse, die für den Erhalt solcher Ereignisse registriert ist, das Ereignis **ITAddressEvent** generiert. Dieses Ereignis gibt an, dass sich der Status der Adressen geändert hat. Der Status ändert sich in AS_OUTOFSERVICE. Bei erfolgreicher Wiederherstellung der Verbindung werden keine Ereignisse generiert. Es werden jedoch alle TAPI 3-Adressen wiederhergestellt.

TAPI 2.x-Referenz

TAPI-Funktionen

In diesem Abschnitt werden die einzelnen Funktionen in TAPI 2.x beschrieben, die vom IP Office-TAPI-Treiber unterstützt werden. Es werden alle speziellen Vorgehensweisen und Beschränkungen der Funktionen bei der Verwendung mit IP Office erläutert.

- lineAddToConference
- lineAnswer
- lineBlindTransfer
- lineCompleteTransfer
- lineConfigDialog
- lineClose
- lineDeallocateCall
- lineDevSpecific
- lineDial
- lineDrop
- lineGenerateDigits
- lineGenerateTone
- lineGetAddressCaps
- lineGetAddressID
- lineGetAddressStatus
- lineGetAppPriority
- lineGetCallInfo
- lineGetCallStatus
- lineGetDevCaps
- lineGetID
- lineGetLineDevStatus
- lineHandoff
- lineHold
- lineInitializeEx
- lineMakeCall
- lineMonitorDigits
- lineMonitorTone
- lineNegotiateAPIVersion
- lineOpen
- linePark
- lineRedirect
- lineRemoveFromConference
- lineSetAppPriority
- lineSetAppSpecific
- lineSetCallPrivilege
- lineSetStatusMessages
- lineSetupTransfer
- lineShutdown
- lineSwapHold
- lineUnhold
- lineUnpark

lineAddToConference

Schaltet den Anruf in das Konferenzgespräch.

```
LONG
WINAPI
lineAddToConference(
HCALL hConfCall,
HCALL hConsultCall
);
```

lineAnswer

Beantwortet einen an der Anwendung wartenden Anruf.

```
LONG
WINAPI
lineAnswer(
HCALL hCall,
LPCSTR lpsUserUserInfo,
DWORD dwSize
);
```

Hinweis: UserUserInfo wird nicht unterstützt und somit ignoriert.

lineBlindTransfer

Diese Funktion dient zur Weiterleitung eines aktiven Anrufs an einen dritten Teilnehmer. Der Ländercode wird ignoriert.

```
LONG
WINAPI
lineBlindTransfer(
HCALL hCall,
LPCSTR lpszDestAddress,
DWORD dwCountryCode
);
```

lineCompleteTransfer

Mit dieser Funktion können Sie eine Vermittlung oder die Einleitung eines Konferenzgesprächs abschließen. Diese Funktion sollte eine Anruf-ID zurückgeben, gibt jedoch stets 0 aus.

```
LONG
WINAPI
lineCompleteTransfer(
HCALL hCall,
HCALL hConsultCall,
LPHCALL lphConfCall,
DWORD dwTransferMode
);
```

lineConfigDialog

Hiermit können Sie dasselbe Dialogfeld zur Konfiguration des TAPI-Dienstanbieters aufrufen wie unter **Systemsteuerung****Telefon- und Modemoptionen** (oder **Telefonie**). Der Parameter **lpszDeviceClass** wird ignoriert.

```
LONG

WINAPI

lineConfigDialog(

DWORD dwDeviceID,
HWND hwndOwner,
LPCSTR lpszDeviceClass
);
```

lineDeallocateCall

Hiermit können Sie die Zuordnung von Ressourcen zu einem Anruf aufheben. Diese Funktion sollte aufgerufen werden, wenn der Anruf inaktiv ist.

```
LONG

WINAPI

lineDeallocateCall(

HCALL hCall
);
```

lineDevSpecific

lineDevSpecific

TAPI bietet erweiterte Funktionalität durch die Funktion **lineDevSpecific**.

Beachten Sie, dass diese nur mit der lizenzierten Version des TAPI-Treibers verfügbar ist.

Mit Hilfe der Funktion **lineDevSpecific** wird ein Puffer unverändert an den TSP übermittelt, wo dieser wie gerätespezifische Befehle interpretiert wird. Es folgt eine Beschreibung der Befehlstypen:

```
LONG

WINAPI

lineDevSpecific(

HLINE hLine,
DWORD dwAddressID,
HCALL hCall,
LPVOID lpParams,
DWORD dwSize
);
```

Das Login-Protokoll

Um einen ACD-Agenten für die überwachte Leitung zu schalten, setzen Sie das erste Byte im Puffer auf **8**. Die folgenden Byte müssen eine Zeichenfolge für die Nebenstelle enthalten, die geschaltet wird. Mit dem folgenden Puffer wird der Agent 218 für die Leitung geschaltet, für die **lineDevSpecific** aufgerufen wird:

```
unsigned char buf[6];

int len = 6;

buf[0] = 8; // Constant that means Login

buf[1] = '2';

buf[2] = '1';

buf[3] = '8';

buf[4] = 0; // Don't forget the null terminator
```

Abmelden

Sie können sich abmelden, indem Sie folgenden Puffer an die Funktion **DevSpecific** übergeben:

```
unsigned char buf[3];

int len = 3;

buf[0] = 9; // Constant that means Shortcode

buf[1] = 47; // Constant that means Log off

buf[2] = 0; // Don't forget the null terminator
```

Umleitungsziel

Um das Ziel für umgeleitete Anrufe festzulegen, senden Sie 9 im ersten Byte, 6 im zweiten Byte und in den folgenden Byte eine Zeichenfolge mit der Nebenstelle des Umleitungsziels. Wenn Sie beispielsweise die Nebenstelle 236 als Umleitungsziel festlegen möchten, senden Sie folgenden Puffer:

```
unsigned char buf[6];
int len = 6;
buf[0] = 9; // The first two bytes are devspecific constants
buf[1] = 6;
buf[2] = '2';
buf[3] = '3';
buf[4] = '6';
buf[5] = 0; // Don't forget the null terminator
```

LED "Nachricht wartet"

Bei einigen Telefonen leuchtet eine LED, wenn neue VoiceMail-Nachrichten für den Benutzer vorliegen. Die Anzahl der neuen Nachrichten kann durch einen gerätespezifischen Befehl gesteuert werden. Bei 0 neuen Nachrichten leuchtet die LED nicht. Ab einer neuen Nachricht leuchtet die LED. Senden Sie den folgenden Puffer an **lineDevSpecific**:

```
unsigned char buf[21];
int len = 21;
buf[0] = 9; // Shortcode
buf[1] = 73; // Set MWL
sprintf(&(buffer[2]), ";Mailbox Msgs=%d", num);
// Where num is the number of messages
```

- Beachten Sie, dass die LED auch vom IP Office-Server oder anderen IP Office-Anwendungen gesteuert werden kann.

Einstellungen für die Weiterleitung (Umleitung)

Die folgenden Konstanten helfen beim Ein- und Ausschalten von Umleitungsfunktionen:

```
const unsigned char ForwardAllOn = 0;
const unsigned char ForwardAllOff = 1;
const unsigned char ForwardBusyOn = 2;
const unsigned char ForwardBusyOff = 3;
const unsigned char ForwardNoAnswerOn = 4;
const unsigned char ForwardNoAnswerOff = 5;
const unsigned char DoNotDisturbOn = 7;
const unsigned char DoNotDisturbOff = 8;
```

Ein Puffer, der eine dieser Konstanten verwendet, sollte drei Byte lang sein und mit 9 beginnen. Mit dem folgenden Code wird die Leitung beispielsweise in den Modus **Nicht stören** geschaltet.

```
unsigned char buf[3];
int len = 3;
buf[0] = 9;
buf[1] = DoNotDisturbOn;
buf[2] = 0;
```

Sammelanschluss aktivieren/deaktivieren

Senden Sie folgenden Puffer, um die Zugehörigkeit des Benutzers zum Sammelanschluss mit der Nebenstelle *groupnum* zu aktivieren.

```
unsigned char buf[10];
int len = 10;
buf[0] = 9;
buf[1] = 76;
sprintf((char*)&buf[2], "%d", groupnum);
```

Senden Sie folgenden Puffer, um die Zugehörigkeit des Benutzers zum Sammelanschluss mit der Nebenstelle *groupnum* zu deaktivieren.

```
unsigned char buf[10];
int len = 10;
buf[0] = 9;
buf[1] = 77;
sprintf((char*)&buf[2], "%d", groupnum);
```

In beiden Fällen können Sie *alle* Zugehörigkeiten aktivieren bzw. deaktivieren, indem Sie die Nebenstellennummer durch 0 ersetzen (z.B. buf[2] = 0).

- Beachten Sie, dass Sie die Zugehörigkeit eines Benutzers nur für Sammelanschlüsse aktivieren bzw. deaktivieren können, für die der Benutzer in Manager als Mitglied konfiguriert wurde.

Aufschalten

Senden Sie den folgenden Puffer, um sich in ein anderes Gespräch einzuschalten. Der Gesprächsteilnehmer, in dessen Gespräch Sie sich einschalten, wird durch die Ganzzahl *extnnum* dargestellt.

```
unsigned char buf[10];
int len = 10;
buf[0] = 9; // Shortcode
buf[1] = 83; // Intrude
sprintf((char*)&buf[2], "%d", extnnum);
```

Abhören

Senden Sie den folgenden Puffer, um ein anderes Gespräch abzuhören. Der Gesprächsteilnehmer, dessen Gespräch Sie abhören möchten, wird durch die Ganzzahl *extnnum* dargestellt.

```
unsigned char buf[10];
int len = 10;
buf[0] = 9; // Shortcode
buf[1] = 100; // Listen
sprintf((char*)&buf[2], "%d", extnnum);
```

lineDial

Mit dieser Funktion können Sie während eines Telefongesprächs eine Nummer wählen. Sie kann als Teil einer überwachten Weiterleitung verwendet werden (siehe lineSetupTransfer). Der Ländercode wird ignoriert.

```
LONG

WINAPI

lineDial(

HCALL hCall,
LPCSTR lpszDestAddress,
DWORD dwCountryCode
);
```

lineDrop

Mit dieser Funktion legen Sie auf bzw. beenden ein Gespräch. **UserUserInfo** wird nicht unterstützt und somit ignoriert.

```
LONG

WINAPI

lineDrop(

HCALL hCall,
LPCSTR lpsUserUserInfo,
DWORD dwSize
);
```

lineGenerateDigits

Rufen Sie diese Funktion auf, um bei einem Anruf DTMF-Ziffern zu generieren. Hierfür muss der Benutzer kein Wave-Benutzer sein und der Wave-Treiber muss nicht in den Anruf involviert sein. Wenn der Generierungsvorgang abgeschlossen ist, wird eine LINE_GENERATE-Meldung an die Anwendung gesendet. Der einzige unterstützte Wert für **dwDigitMode** ist LINEDIGITMODE_DTMF.

```
LONG

WINAPI

lineGenerateDigits(

HCALL hCall,
DWORD dwDigitMode,
LPCSTR lpszDigits,
DWORD dwDuration
);
```

lineGenerateTone

Mit dieser Funktion können Sie einen Signalton in der Leitung generieren. Bei der Leitung muss es sich um einen Wave-Benutzer handeln und der Wave-Treiber muss in den Anruf involviert sein. Der einzige unterstützte Wert für **dwToneMode** ist LINETONEMODE_BEEP. Da benutzerdefinierte Signaltöne nicht unterstützt werden, sollten Sie **dwNumTones** auf 0 setzen.

```
LONG

WINAPI

lineGenerateTone(

HCALL hCall,
DWORD dwToneMode,
DWORD dwDuration,
DWORD dwNumTones,
LPLINEGENERATETONE const lpTones
);
```

lineGetAddressCaps

Dient zum Abrufen der Telefoniefunktionen einer bestimmten Adresse für eine bestimmte Leitung. Die Funktionen werden in der Struktur LINEADDRESSCAPS zurückgegeben. Nähere Informationen hierzu finden Sie im Abschnitt zu TAPI-Strukturen unter *LINEADDRESSCAPS*.

IP Office-Leitungen verfügen stets über eine einzige Adresse.

```
LONG

WINAPI

lineGetAddressCaps(

HLINEAPP hLineApp,
DWORD dwDeviceID,
DWORD dwAddressID,
DWORD dwAPIVersion,
DWORD dwExtVersion,
LPLINEADDRESSCAPS lpAddressCaps
);
```

lineGetAddressID

Diese Funktion ermöglicht die Zuordnung der Telefonnummer (Adresse) einer Leitung zu ihrer *dwAddressID* im Bereich 0 bis zu der in LINEDEVCAPS zurückgegebenen Anzahl an Adressen minus 1. Wenn **dwNumAddresses** in LINEDEVCAPS den Wert 1 aufweist, gibt diese Funktion für den Wert DWORD, auf den **lpdwAddressID** verweist, stets 0 zurück.

```
LONG

WINAPI

lineGetAddressID(

HLINE hLine,
LPDWORD lpdwAddressID,
DWORD dwAddressMode,
LPCSTR lpsAddress,
DWORD dwSize
);
```

lineGetAddressStatus

Diese Funktion ermöglicht einer Anwendung, den aktuellen Status der angegebenen Adresse abzufragen. Nähere Informationen hierzu finden Sie im Abschnitt zu den TAPI-Strukturen unter *LINEADDRESSSTATUS*.

```
LONG

WINAPI

lineGetAddressStatus(

HLINE hLine,
DWORD dwAddressID,
LPLINEADDRESSSTATUS lpAddressStatus
);
```

lineGetAppPriority

Dient zum Abrufen der Anwendungspriorität.

```
LONG

WINAPI

lineGetAppPriority(

LPCSTR lpszAppFilename,
DWORD dwMediaMode,
LPLINEEXTENSIONID lpExtensionID,
DWORD dwRequestMode,
LPVARSTRING lpExtensionName,
LPDWORD lpdwPriority
);
```

lineGetCallInfo

Hiermit können feste Informationen über den angegebenen Anruf abgerufen werden. Nähere Informationen finden Sie in der Erläuterung zur Struktur LINECALLINFO.

```
LONG

WINAPI

lineGetCallInfo(

HCALL hCall,
LPLINECALLINFO lpCallInfo
);
```

lineGetCallStatus

Mit dieser Funktion wird eine LINECALLSTATUS-Struktur im Zusammenhang mit einem vorhandenen Anruf abgerufen. Nähere Informationen hierzu finden Sie im Abschnitt zu den TAPI-Strukturen unter *LINECALLSTATUS*.

```
LONG

WINAPI

lineGetCallStatus(

HCALL hCall,
LPLINECALLSTATUS lpCallStatus
);
```

lineGetDevCaps

Mit dieser Funktion können Sie die Struktur LINEDEVCAPS abrufen. Nähere Informationen hierzu finden Sie im Abschnitt zu den TAPI-Strukturen unter *LINEDEVCAPS*.

```
LONG

WINAPI

lineGetDevCaps(

HLINEAPP hLineApp,
DWORD dwDeviceID,
DWORD dwAPIVersion,
DWORD dwExtVersion,
LPLINEDEVCAPS lpLineDevCaps
);
```

lineGetID

Dient zum Abrufen der ID für eine Leitung, wenn **dwSelect** auf LINECALLSELECT_LINE gesetzt ist.

```
LONG

WINAPI

lineGetID(

HLINE hLine,
DWORD dwAddressID,
HCALL hCall,
DWORD dwSelect,
LPVARSTRING lpDeviceID,
LPCSTR lpszDeviceClass
);
```

lineGetLineDevStatus

Die Funktion **lineGetLineDevStatus** gibt einen gerätespezifischen Puffer zurück. Der gerätespezifische Puffer enthält folgende Informationen:

```
LONG

WINAPI

lineGetLineDevStatus(

HLINE hLine,
LPLINEDEVSTATUS lpLineDevStatus
);
```

Byte	Enthält	Kommentar
0..n	Telefonnebenstelle	Die überwachte Nummer der Leitung als Zeichenfolge (z.B. "217").
n+1	0	Nullabschlusszeichen für die obige Zeichenfolge.
n+2	Rufweiterleitung bei Besetzt	1, wenn **Rufweiterleitung bei Besetzt** für das Telefon aktiviert ist, andernfalls 0.
n+3	Rufweiterleitung nach Zeit	1, wenn **Rufweiterleitung nach Zeit** für das Telefon aktiviert ist, andernfalls 0.
n+4	Rufweiterleitung sofort	1, wenn die sofortige Rufweiterleitung für das Telefon aktiviert ist.
n+5	Rufweiterleitung für Sammelanschlussanrufe (Flag)	1, wenn die Rufweiterleitung für Sammelanschlussanrufe für das Telefon aktiviert ist.
n+6	Nicht stören	1, wenn **Nicht stören** für das Telefon aktiviert ist.
n+7	Sperre für ausgehende Anrufe (Flag)	1, wenn das Telefon für externe Anrufe gesperrt ist.
n+8	Anklopfen ein (Flag)	1, wenn die Anklopffunktion für dieses Telefon aktiviert ist.
n+9	VoiceMail ein (Flag)	1, wenn die VoiceMail-Funktion für dieses Telefon aktiviert ist.
n+10	VoiceMail-Rückruf (Flag)	1, wenn die VoiceMail-Rückruffunktion für dieses Telefon aktiviert ist.
n+11	Anzahl der gelesenen VoiceMail-Nachrichten	Die Anzahl der gelesenen Nachrichten.
n+12	Anzahl der nicht gelesenen VoiceMail-Nachrichten	Die Anzahl der neuen VoiceMail-Nachrichten für den Benutzer.
n+13	Ruffolge extern (Anzahl)	Die Ruffolge für externe Anrufe.
n+14	Ruffolge intern (Anzahl)	Die Ruffolge für interne Anrufe.
n+15	Ruffolge bei Rückruf (Anzahl)	Die Ruffolge für Rückrufe.

n+16	Zeitüberschreitung bei keiner Antwort (Zeitspanne)	Gibt an, wie lange das Telefon klingelt (Sekunden), bevor die *keine Antwort*-Aktion ausgeführt wird, z.B. Weiterleitung oder Umleitung an VoiceMail.
n+17	Nachbearbeitungszeit (Zeitspanne)	Der Zeitraum in Sekunden, in dem das Telefon nach einem Anruf keine Anrufe empfangen kann.
n+18	Aufschalten erlaubt (Flag)	1, wenn die Aufschaltung mit diesem Telefon möglich ist.
n+19	Aufschalten nicht erlaubt (Flag)	1, wenn eine Aufschaltung bei diesem Telefon nicht möglich ist.
n+20	Geheimnummer (Flag)	1, wenn dieser Benutzer nicht im internen Verzeichnis vorhanden ist.
n+21	Login erzwingen (Flag)	Der Benutzer ist beim Systemstart abgemeldet und muss sich somit anmelden.
n+22	Verrechnungscode erzwingen (Flag)	1, wenn für externe Anrufe mit diesem Telefon ein gültiger Verrechnungscode angegeben werden muss.
n+23	Login-Code (Flag)	1, wenn für diesen Benutzer ein Login-Code konfiguriert ist.
n+24	Systemtelefon (Flag)	1, wenn es sich um ein Systemtelefon handelt.
n+25	Abwesenheitsnachricht (ID)	Die ID der Abwesenheitsnachricht.
n+26	Abwesenheitsnachricht eingerichtet (Flag)	1, wenn die Abwesenheitsnachricht mit der im vorigen Feld angegebenen ID am Telefon angezeigt wird.
n+27	VoiceMail als E-Mail (Modus)	1, wenn der Modus **VoiceMail als E-Mail** aktiviert ist.
n+28..m	Nst	Die Benutzernebenstelle, die sich möglicherweise von der Telefonnebenstelle unterscheidet.
m+1	0	Nullabschlusszeichen für die obige Zeichenfolge (Nst).
m+2..p	Länderkennung	Die Länderkennung des Benutzers.
p+1	0	Das Nullabschlusszeichen für die Länderkennung.
p+2..q	Weiterleitungsziel	Die Nummer, die für dieses Telefon als Weiterleitungsziel festgelegt ist.
q+1	0	Das Nullabschlusszeichen für das oben angegebene Ziel.
q+2..r	Umleitungsnummer	Alle Anrufe werden an diese Nummer umgeleitet.
r+1	0	

r+2..s	Abwesenheitstext	Der für dieses Telefon definierte Abwesenheitstext.
s+1	0	
s+2..t	Nicht stören-Ausnahmeliste	Eine Liste mit Nummern, deren Anrufe für dieses Telefon durchgestellt werden, während es sich im Status **Nicht stören** befindet. Jede Nummer besteht aus einer Zeichenfolge mit Nullabschlusszeichen. Die letzte Nummer in der Liste endet mit zwei Nullen (das Feld **t+1** enthält die zweite Null). Wenn diese Liste leer ist, enthalten die Daten lediglich eine Null (im Feld **t+1**).
t+1	0	
t+2..u	Weiterleitungsnummer bei Besetzt	Die Nummer, an die Anrufe weitergeleitet werden, wenn dieses Telefon besetzt ist.
u+1	0	
u+2	Benutzerpriorität	Diese Priorität wird allen von diesem Benutzer getätigten Anrufen zugewiesen.
u+3	Gruppenmitgliedschaft	Dieses Byte enthält die Anzahl der Gruppen, für die der Benutzer zurzeit als Mitglied aktiviert ist.
u+4	Gruppen außerhalb des Zeitprofils	Anzahl der Gruppen, in denen der Benutzer Mitglied ist und die sich zurzeit außerhalb ihres Zeitprofils befinden.
u+5	Deaktivierte Gruppen	Anzahl der Gruppen, die für den Benutzer zurzeit deaktiviert sind.
u+6	Gruppen außer Betrieb	Anzahl der Gruppen, in denen der Benutzer Mitglied ist und die zurzeit außer Betrieb sind.
u+7	Gruppen im Nachtbetrieb	Anzahl der Gruppen, in denen der Benutzer Mitglied ist und die zurzeit im Nachtbetrieb sind.

lineHandoff

Mit der Funktion **lineHandoff** wird der angegebene Anruf an eine andere Anwendung übergeben.

```
LONG

WINAPI

lineHandoff(

HCALL hCall,
LPCSTR lpszFileName,
DWORD dwMediaMode
);
```

lineHold

Mit dieser Funktion wird ein aktiver Anruf gehalten.

```
LONG

WINAPI

lineHold(

HCALL hCall
);
```

lineInitializeEx

Dies ist die erste TAPI-Funktion, die zur Initialisierung von TAPI aufgerufen werden muss. Der Parameter **lpdwAPIVersion** muss mindestens auf **0x00020000** gesetzt werden.

```
LONG

WINAPI

lineInitializeEx(

LPHLINEAPP lphLineApp,
HINSTANCE hInstance,
LINECALLBACK lpfnCallback,
LPCSTR lpszFriendlyAppName,
LPDWORD lpdwNumDevs,
LPDWORD lpdwAPIVersion,
LPLINEINITIALIZEEXPARAMS lpLineInitializeExParams
);
```

lineMakeCall

Mit dieser Funktion wird ein Anruf getätigt. Nähere Informationen finden Sie am Ende des Kapitels zu den TAPI-Funktionen im Abschnitt zu den Anrufparametern.

```
LONG

WINAPI

lineMakeCall(

HLINE hLine,
LPHCALL lphCall,
LPCSTR lpszDestAddress,
DWORD dwCountryCode,
LPLINECALLPARAMS const lpCallParams
);
```

lineMonitorDigits

Rufen Sie diese Funktion auf, um die Erkennung von DTMF-Ziffern zu aktivieren. Diese Funktion kann nur verwendet werden, wenn der IP Office-Wave-Treiber in den Anruf involviert ist und der Benutzer ein Wave-Benutzer ist (siehe Abschnitt *Wave-Benutzer*). Die Erkennung erfolgt durch die Analyse von Medienstichproben im Wave-Treiber. Wenn ein DTMF-Ton erkannt wird, wird eine LINE_MONITORDIGITS-Meldung an die Anwendung gesendet. **dwDigitModes** kann auf LINEDIGITMODE_DTMF und/oder auf LINEDIGITMODE_DTMFEND gesetzt werden. Rufen Sie die Funktion **lineMonitorDigits** mit dem Wert 0 für **dwDigitMode** auf, um die Erkennung von DTMF-Ziffern abzubrechen.

```
LONG

WINAPI

lineMonitorDigits(

HCALL hCall,
DWORD dwDigitModes
);
```

lineMonitorTone

Damit diese Funktion ausgeführt werden kann, muss wie bei der vorigen Funktion der Wave-Treiber in den Anruf involviert sein. Darüber hinaus dient sie lediglich zur Erkennung von Momenten ohne Ton. Die Frequenzen in der Struktur LINEMONITORTONE, auf die **lpToneList** verweist, müssen alle auf 0 gesetzt werden. Bei Erkennung eines Moments ohne Ton wird eine LINE_MONITORTONE-Meldung an die Anwendung gesendet. Rufen Sie die Funktion **lineMonitorTone** mit dem Wert NULL für **lpToneList** auf, um die Erkennung von Momenten ohne Ton abzubrechen.

```
LONG

WINAPI

lineMonitorTone(

HCALL hCall,
LPLINEMONITORTONE const lpToneList,
DWORD dwNumEntries
);
```

lineNegotiateAPIVersion

Rufen Sie diese Funktion direkt nach **lineInitializeEx** auf, um sicherzustellen, dass die richtigen TAPI-Meldungen an Ihre Anwendung gesendet werden. Diese Funktion muss für jede von Ihrer Anwendung verwendete Leitung aufgerufen werden.

```
LONG

WINAPI

lineNegotiateAPIVersion(

HLINEAPP hLineApp,
DWORD dwDeviceID,
DWORD dwAPILowVersion,
DWORD dwAPIHighVersion,
LPDWORD lpdwAPIVersion,
LPLINEEXTENSIONID lpExtensionID
);
```

lineOpen

Diese Funktion dient zum Öffnen einer Leitung.

Bei ISDN/T1-Amtsleitungen muss **dwMediaModes** auf LINEMEDIAMODE_INTERACTIVEVOICE und bei analogen Amtsleitungen auf LINEMEDIAMODE_UNKNOWN gesetzt werden. Sie können auch beide Werte angeben, damit Anrufe beider Leitungstypen verarbeitet werden.

```
LONG

WINAPI

lineOpen(

HLINEAPP hLineApp,
DWORD dwDeviceID,
LPHLINE lphLine,
DWORD dwAPIVersion,
DWORD dwExtVersion,
DWORD dwCallbackInstance,
DWORD dwPrivileges,
DWORD dwMediaModes,
LPLINECALLPARAMS const lpCallParams
);
```

Hinweis: Wenn Sie versuchen, eine Leitung zu öffnen, die mit einem Wave-Benutzer verknüpft ist, in IP Office jedoch keine Wave-Benutzerlizenz installiert wurde, gibt die Funktion **lineOpen** den Wert LINEERR_RESOURCEUNAVAIL zurück. Eine Erläuterung zu Wave-Benutzern finden Sie unter *Wave-Benutzer*.

linePark

Mit dieser Funktion wird ein Anruf geparkt. Es wird nur der Modus LINEPARKMODE_DIRECTED unterstützt. Als Adressen können alphanumerische Zeichenfolgen verwendet werden. Da über ein Telefon jedoch nur Ziffern eingegeben werden können, möchten Sie die Adressen zum Parken von Anrufen möglicherweise auf numerische Zeichenfolgen beschränken. Die vier standardmäßigen Adressen in Phone Manager und eConsole lauten 1, 2, 3 und 4. Es wird empfohlen, diese Zahlen zu verwenden, wenn Sie geparkte Anrufe mit diesen Anwendungen unter Verwendung der Standardkonfiguration entparken möchten.

```
LONG

WINAPI

linePark(

HCALL hCall,
DWORD dwParkMode,
LPCSTR lpszDirAddress,
LPVARSTRING lpNonDirAddress
);
```

lineRedirect

Mit der Funktion **lineRedirect** wird der angegebene Anruf an die angegebene Zieladresse umgeleitet. Der Ländercode wird ignoriert.

```
LONG

WINAPI

lineRedirect(

HCALL hCall,
LPCSTR lpszDestAddress,
DWORD dwCountryCode
);
```

lineRemoveFromConference

Diese Funktion dient zur Entfernung eines Anrufs aus einem Konferenzgespräch.

```
LONG

WINAPI

lineRemoveFromConference(

HCALL hCall
);
```

lineSetAppPriority

Rufen Sie diese Funktion auf, um Ihre Anwendungspriorität anzugeben.

```
LONG

WINAPI

lineSetAppPriority(

LPCSTR lpszAppFilename,
DWORD dwMediaMode,
LPLINEEXTENSIONID lpExtensionID,
DWORD dwRequestMode,
LPCSTR lpszExtensionName,
DWORD dwPriority
);
```

lineSetAppSpecific

Diese Funktion ermöglicht die Einrichtung des anwendungsspezifischen Feldes für den Informationsdatensatz des angegebenen Anrufs in einer Anwendung.

```
LONG

WINAPI

lineSetAppSpecific(

HCALL hCall,
DWORD dwAppSpecific
);
```

lineSetCallPrivilege

Rufen Sie diese Funktion auf, um die Anwendungs-Eigentumsrechte für einen bestimmten Anruf zu ändern.

```
LONG

WINAPI

lineSetCallPrivilege(

HCALL hCall,
DWORD dwCallPrivilege
);
```

lineSetStatusMessages

Mit dieser Funktion kann die Anwendung angeben, welche Benachrichtigungsmeldungen sie erfordert. In der Regel ist **dwLineStates** auf LINEDEVSTATE_ALL und **dwAddressStates** auf LINEADDRESSSTATE_ALL gesetzt.

```
LONG

WINAPI

lineSetStatusMessages(

HLINE hLine,
DWORD dwLineStates,
DWORD dwAddressStates
);
```

lineSetupTransfer

Diese Funktion dient zur Erstellung eines Rückfragerufs zur Durchführung einer überwachten Vermittlung. Der zu vermittelnde Anruf muss bereits vorhanden sein. Beim Aufruf dieser Funktion muss der Anruf entweder aktiv oder auf Halten gesetzt sein. Ein aktiver Anruf wird durch diese Funktion auf Halten gesetzt. Rufen Sie die Funktion **lineDial** auf, um den Teilnehmer anzurufen, an den der Anruf vermittelt werden soll. Schließen Sie die Vermittlung dann mit Hilfe der Funktion **lineCompleteTransfer** ab.

```
LONG

WINAPI

lineSetupTransfer(

HCALL hCall,
LPHCALL lphConsultCall,
LPLINECALLPARAMS const lpCallParams
);
```

lineShutdown

Hiermit beenden Sie die Verwendung der Funktionen für die TAPI-Leitung. In der Regel wird diese Funktion beim Herunterfahren Ihrer Anwendung aufgerufen.

```
LONG

WINAPI

lineShutdown(

HLINEAPP hLineApp
);
```

lineSwapHold

Mit dieser Funktion wird der aktuelle aktive Anruf auf Halten gesetzt und der gehaltene Anruf empfangen.

```
LONG

WINAPI

lineSwapHold(

HCALL hActiveCall,
HCALL hHeldCall
);
```

lineUnhold

Mit dieser Funktion wird ein gehaltener Anruf herangeholt. Wenn über die Leitung ein weiterer Gesprächspartner angerufen wird oder ein weiterer Anruf aktiv ist, wird bei Aufruf dieser Funktion der gestartete/aktive Anruf verworfen, bevor der gehaltene Anruf abgerufen wird.

```
LONG

WINAPI

lineUnhold(

HCALL hCall
);
```

lineUnpark

Diese Funktion dient zum Abrufen eines geparkten Anrufs. **dwAddressID** sollte auf **0** gesetzt sein, da IP Office-Leitungen lediglich über eine Adresse verfügen. **lpszDestAddress** sollte auf denselben Wert gesetzt werden, der auch zum Parken des Anrufs verwendet wurde (siehe **linePark**).

```
LONG

WINAPI

lineUnpark(

HLINE hLine,
DWORD dwAddressID,
LPHCALL lphCall,
LPCSTR lpszDestAddress
);
```

TAPI-Strukturen

LINEADDRESSCAPS

Diese Struktur wird von der Funktion **lineGetAddressCaps** zurückgegeben. In der folgenden Tabelle sind die Werte aufgeführt, die für Leitungen in Verbindung mit dem IP Office-TAPI-Treiber zurückgegeben werden.

- **Hinweis:** Die Tabelle enthält nicht alle Details dieser Struktur. Eine ausführliche Erläuterung der möglichen Werte für LINEADDRESSCAPS finden Sie in der Dokumentation zu Microsoft TAPI.

Element	Beschreibung/Wert
dwLineDeviceID	Die ID der Leitung, mit der diese Adresse verbunden ist.
dwDevSpecificSize	Es werden keine zusätzlichen gerätespezifischen Informationen übergeben.
dwDevSpecificOffset	0
dwAddressSharing	LINEADDRESSSHARING_PRIVATE
dwAddressStates	0
dwCallInfoStates	Dient zur Rückgabe der möglichen Anrufinformationsstatus: LINECALLINFOSTATE_CALLID LINECALLINFOSTATE_RELATEDCALLID LINECALLINFOSTATE_NUMOWNERINCR LINECALLINFOSTATE_NUMOWNEDECR LINECALLINFOSTATE_NUMMONITORS LINECALLINFOSTATE_CALLERID LINECALLINFOSTATE_CALLEDID LINECALLINFOSTATE_REDIRECTIONID LINECALLINFOSTATE_REDIRECTINGID LINECALLINFOSTATE_DISPLAY LINECALLINFOSTATE_MONITORMODES LINECALLINFOSTATE_CALLDATA
dwCallerIDFlags	Dient zur Rückgabe der möglichen Markierungen für die Rufnummer des Anrufers: LINECALLPARTYID_BLOCKED LINECALLPARTYID_OUTOFAREA LINECALLPARTYID_NAME LINECALLPARTYID_ADDRESS LINECALLPARTYID_UNKNOWN LINECALLPARTYID_UNAVAIL

dwCalledIDFlags	Dient zur Rückgabe der möglichen Markierungen für die Rufnummer des angerufenen Teilnehmers: LINECALLPARTYID_BLOCKED LINECALLPARTYID_OUTOFAREA LINECALLPARTYID_NAME LINECALLPARTYID_ADDRESS LINECALLPARTYID_UNKNOWN LINECALLPARTYID_UNAVAIL
dwConnectedIDFlags	Dient zur Rückgabe der möglichen Verbindungs-ID-Markierungen: LINECALLPARTYID_NAME LINECALLPARTYID_UNKNOWN LINECALLPARTYID_UNAVAIL
dwRedirectionIDFlags	Dient zur Rückgabe der möglichen Markierungen für die Rufnummer des Umleitungsziels: LINECALLPARTYID_BLOCKED LINECALLPARTYID_OUTOFAREA LINECALLPARTYID_NAME LINECALLPARTYID_ADDRESS LINECALLPARTYID_UNKNOWN LINECALLPARTYID_UNAVAIL
dwRedirectingIDFlags	Dient zur Rückgabe der möglichen Markierungen für die Rufnummer, von der die Umleitung ausgeht: LINECALLPARTYID_BLOCKED LINECALLPARTYID_OUTOFAREA LINECALLPARTYID_NAME LINECALLPARTYID_ADDRESS LINECALLPARTYID_UNKNOWN LINECALLPARTYID_UNAVAIL

dwCallStates	Dient zur Rückgabe der möglichen Anrufstatus: LINECALLSTATE_IDLE (Anruf nicht mehr vorhanden) LINECALLSTATE_OFFERING (neuen Anruf empfangen) LINECALLSTATE_ACCEPTED (Anruf wurde von einer Anwendung angefordert) LINECALLSTATE_DIALTONE (Anrufer hört Wählton) LINECALLSTATE_DIALING (Vermittlung empfängt Wählinformationen) LINECALLSTATE_RINGBACK (Anrufer hört das Klingelzeichen) LINECALLSTATE_BUSY (Anrufer hört das Besetztzeichen) LINECALLSTATE_CONNECTED (Verbindung vollständig hergestellt) LINECALLSTATE_PROCEEDING (Wählvorgang abgeschlossen, Anruf noch nicht durchgestellt) LINECALLSTATE_ONHOLD (Anruf wurde auf Halten gesetzt) LINECALLSTATE_CONFERENCED (Anruf in ein Konferenzgespräch geschaltet) LINECALLSTATE_ONHOLDPENDCONF (Anruf vor Konferenzschaltung auf Halten gesetzt) LINECALLSTATE_ONHOLDPENDTRANSFER (Anruf vor Vermittlung auf Halten gesetzt) LINECALLSTATE_DISCONNECTED (Verbindung wurde vom anderen Gesprächsteilnehmer getrennt) LINECALLSTATE_UNKNOWN (Anrufstatus unbekannt)
dwDialToneModes	Dient zur Rückgabe des möglichen Wähltonmodus: LINEDIALTONEMODE_UNAVAIL
dwBusyModes	Dient zur Rückgabe der möglichen Besetzt-Modi: LINEBUSYMODE_UNAVAIL
dwSpecialInfo	Dient zur Rückgabe der möglichen Spezialinformationen aus LINESPECIALINFO_UNAVAIL

dwDisconnectModes	Dient zur Rückgabe der möglichen Trennen-Modi: LINEDISCONNECTMODE_NORMAL LINEDISCONNECTMODE_REJECT LINEDISCONNECTMODE_PICKUP LINEDISCONNECTMODE_FORWARDED LINEDISCONNECTMODE_BUSY LINEDISCONNECTMODE_NOANSWER LINEDISCONNECTMODE_BADADDRESS LINEDISCONNECTMODE_UNREACHABLE LINEDISCONNECTMODE_CONGESTION LINEDISCONNECTMODE_INCOMPATIBLE LINEDISCONNECTMODE_UNAVAIL LINEDISCONNECTMODE_NODIALTONE LINEDISCONNECTMODE_QOSUNAVAIL LINEDISCONNECTMODE_BLOCKED LINEDISCONNECTMODE_DONOTDISTURB
dwMaxNumActiveCalls	Die maximale Anzahl an aktiven Anrufen: 1
dwMaxNumOnHoldCalls	Die maximale Anzahl an gehaltenen Anrufen: 9
dwMaxNumOnHoldPendingCalls	Die maximale Anzahl an ausstehenden gehaltenen Anrufen: 9
dwMaxNumConference	Die maximale Anzahl an Konferenzgesprächen: 9
dwMaxNumTransConf	Die maximale Anzahl an vermittelten Konferenzgesprächen: 9
dwAddrCapFlags	Dient zur Rückgabe der möglichen Markierungen für den Adresswert: LINEADDRCAPFLAGS_FWDNUMRINGS LINEADDRCAPFLAGS_DIALED LINEADDRCAPFLAGS_TRANSFERHELD LINEADDRCAPFLAGS_TRANSFERMAKE LINEADDRCAPFLAGS_CONFERENCEHELD LINEADDRCAPFLAGS_CONFERENCEMAKE LINEADDRCAPFLAGS_FWDSTATUSVALID

dwCallFeatures	Dient zur Rückgabe der möglichen Anruffunktionen: LINECALLFEATURE_ADDTOCONF LINECALLFEATURE_ANSWER LINECALLFEATURE_BLINDTRANSFER LINECALLFEATURE_COMPLETETRANSF LINECALLFEATURE_DIAL LINECALLFEATURE_DROP LINECALLFEATURE_GENERATEDIGITS LINECALLFEATURE_HOLD LINECALLFEATURE_PARK LINECALLFEATURE_REDIRECT LINECALLFEATURE_REMOVEFROMCONF LINECALLFEATURE_SETUPTRANSFER LINECALLFEATURE_SWAPHOLD LINECALLFEATURE_UNHOLD LINECALLFEATURE_SETCALLDATA
dwRemoveFromConfCaps	Dient zur Rückgabe der möglichen Werte für die Entfernung aus einer Konferenz: LINEREMOVEFROMCONF_ANY
dwRemoveFromConfState	Dient zur Rückgabe des möglichen Status für die Entfernung aus einer Konferenz: LINECALLSTATE_ONHOLD
dwTransferModes	Dient zur Rückgabe der möglichen Vermittlungsmodi: LINETRANSFERMODE_TRANSFER LINETRANSFERMODE_CONFERENCE
dwParkModes	Dient zur Rückgabe des möglichen Parkmodus LINEPARKMODE_DIRECTED
dwForwardModes	Dient zur Rückgabe der möglichen Weiterleitungsmodi: LINE FORWARDMODE_UNCOND LINE FORWARDMODE_UNCONDEXTERNAL LINE FORWARDMODE_UNCONDSPECIFIC LINE FORWARDMODE_BUSY LINE FORWARDMODE_BUSYINTERNAL LINE FORWARDMODE_BUSYEXTERNAL LINE FORWARDMODE_BUSYSPECIFIC LINE FORWARDMODE_NOANSW LINE FORWARDMODE_NOANSWINTERNAL LINE FORWARDMODE_NOANSWEXTERNAL LINE FORWARDMODE_NOANSWSPECIFIC LINE FORWARDMODE_BUSYNA LINE FORWARDMODE_BUSYNAINTERNAL LINE FORWARDMODE_BUSYNAEXTERNAL LINE FORWARDMODE_BUSYNASPECIFIC

dwMaxForwardEntries	Die maximale Anzahl an weitergeleiteten Einträgen: 10
dwMaxSpecificEntries	Die maximale Anzahl an spezifischen Einträgen: 10
dwMinFwdNumRings	Die minimale Anzahl an Klingeltönen bei der Weiterleitung: 1
dwMaxFwdNumRings	Die maximale Anzahl an Klingeltönen bei der Weiterleitung: 99
dwMaxCallCompletions	0
dwCallCompletionConds	0
dwCallCompletionModes	0
dwNumCompletionMessages	0
dwCompletionMsgTextEntrySize	0
dwCompletionMsgTextSize	0
dwCompletionMsgTextOffset	0
dwAddressFeatures	Dient zur Rückgabe der möglichen Adressfunktionen: LINEADDRFEATURE_FORWARD LINEADDRFEATURE_MAKECALL LINEADDRFEATURE_SETUPCONF LINEADDRFEATURE_UNPARK LINEADDRFEATURE_FORWARDFWD LINEADDRFEATURE_FORWARDDND
dwPredictiveAutoTransferStates	0
dwNumCallTreatments	0
dwCallTreatmentListSize	0
dwCallTreatmentListOffset	0
dwDeviceClassesSize	0
dwDeviceClassesOffset	0
dwMaxCallDataSize	Die maximale Anrufdatengröße: 127
dwCallFeatures2	0
dwMaxNoAnswerTimeout	0
dwConnectedModes	0
dwOfferingModes	0
dwAvailableMediaModes	0

LINEADDRESSSTATUS

Diese Struktur wird von der Funktion **lineGetAddressStatus** zurückgegeben.

- **Hinweis:** Die Tabelle enthält nicht alle Details dieser Struktur. Eine ausführliche Erläuterung der möglichen Werte für LINEADDRESSSTATUS finden Sie in der Dokumentation zu Microsoft TAPI.

Element	Beschreibung/Wert
dwNumInUse;	immer 1
dwNumActiveCalls;	Anzahl der aktiven Anrufe
dwNumOnHoldCalls;	immer 0
dwNumOnHoldPendCalls;	immer 0
dwAddressFeatures;	Dient zur Angabe der möglichen Funktionen: LINEADDRFEATURE_MAKECALL LINEADDRFEATURE_SETUPCONF LINEADDRFEATURE_UNPARK
dwNumRingsNoAnswer;	5
dwForwardNumEntries;	immer 0
dwForwardSize; /	immer 0
dwForwardOffset;	immer 0
dwTerminalModesSize;	immer 0
dwTerminalModesOffset;	immer 0
dwDevSpecificSize;	immer 0

LINECALLINFO

Diese Struktur wird von der Funktion **lineGetCallInfo** zurückgegeben.

- **Hinweis:** Die Tabelle enthält nicht alle Details dieser Struktur. Eine ausführliche Erläuterung der möglichen Werte für LINECALLINFO finden Sie in der Dokumentation zu Microsoft TAPI.

Element	Beschreibung/Wert
dwAddressID	immer 0
dwBearerMode	Dient zur Rückgabe des möglichen Dienstmodus: LINEBEARERMODE_VOICE
dwRate	64000
dwMediaMode	Dient zur Rückgabe des möglichen Medienmodus: LINEMEDIAMODE_INTERACTIVEVOICE
dwAppSpecific	Einstellung durch Anwendung
dwCallID	Anrufer-ID
dwCallParamFlags	Dient zur Rückgabe der möglichen Markierung für Anrufparameter: LINECALLPARAMFLAGS_IDLE
dwCallStates	Dient zur Rückgabe der möglichen Anrufstatus: LINECALLSTATE_IDLE LINECALLSTATE_OFFERING LINECALLSTATE_DIALTONE LINECALLSTATE_DIALING LINECALLSTATE_RINGBACK LINECALLSTATE_BUSY LINECALLSTATE_CONNECTED LINECALLSTATE_PROCEEDING LINECALLSTATE_ONHOLD LINECALLSTATE_CONFERENCED LINECALLSTATE_ONHOLDPENDCONF LINECALLSTATE_ONHOLDPENDTRANSFER LINECALLSTATE_DISCONNECTED LINECALLSTATE_UNKNOWN
dwMonitorMediaModes	0
dwCountryCode	0
dwTrunk	0xFFFFFFFF
dwCommentSize	0
dwCommentOffset	0

dwUserUserInfoSize	0
dwUserUserInfoOffset	0
dwHighLevelCompSize	0
dwHighLevelCompOffset	0
dwLowLevelCompSize	0
dwLowLevelCompOffset	0
dwChargingInfoSize	0
dwChargingInfoOffset	0
dwTerminalModesSize	0
dwTerminalModesOffset	0
dwCallDataOffset	0
dwSendingFlowspecSize	0
dwSendingFlowspecOffset	0
dwReceivingFlowspecSize	0
dwReceivingFlowspecOffset	0
dwCallerIDAddressType	0 – nur gültig ab TAPI Version 3.0
dwCalledIDAddressType	0 – nur gültig ab TAPI Version 3.0
dwConnectedIDAddressType	0 – nur gültig ab TAPI Version 3.0
dwRedirectionIDAddressType	0 – nur gültig ab TAPI Version 3.0
dwRedirectingIDAddressType	0 – nur gültig ab TAPI Version 3.0

LINECALLPARAMS

Folgende Parameter der Struktur LINECALLPARAMS können an **lineMakeCall** und **lineSetupTransfer** übergeben werden:

- **Hinweis:** Die Tabelle enthält nicht alle Details dieser Struktur. Eine ausführliche Erläuterung der möglichen Werte für LINECALLPARAMS finden Sie in der Dokumentation zu Microsoft TAPI.

Element	Beschreibung/Wert
dwCallParamFlags	Setzen Sie diese Option für die normale Verwendung auf 0. Wenn Sie die Rufnummer des Anrufers verbergen möchten, geben Sie LINEBEARERMODE_VOICE ein.
dwCalledPartyOffset	Kann zur Einstellung der Rufnummer des angerufenen Teilnehmers verwendet werden.
dwCallingPartyIDOffset	Kann zur Einstellung der Rufnummer des Anrufers verwendet werden.

LINECALLSTATUS

Diese Struktur wird von der Funktion **lineGetCallStatus** zurückgegeben.

- **Hinweis:** Die Tabelle enthält nicht alle Details dieser Struktur. Eine ausführliche Erläuterung der möglichen Werte für LINECALLSTATUS finden Sie in der Dokumentation zu Microsoft TAPI.

Element	Beschreibung/Wert
dwCallState	Gibt einen der folgenden Status zurück: LINECALLSTATE_IDLE (Anruf nicht mehr vorhanden) LINECALLSTATE_OFFERING (neuen Anruf empfangen) LINECALLSTATE_ACCEPTED (Anruf wurde von einer Anwendung angefordert) LINECALLSTATE_DIALTONE (Anrufer hört Wählton) LINECALLSTATE_DIALING (Vermittlung empfängt Wählinformationen) LINECALLSTATE_RINGBACK (Anrufer hört das Klingelzeichen) LINECALLSTATE_BUSY (Anrufer hört das Besetztzeichen) LINECALLSTATE_CONNECTED (Verbindung vollständig hergestellt) LINECALLSTATE_PROCEEDING (Wählvorgang abgeschlossen, Anruf noch nicht durchgestellt) LINECALLSTATE_ONHOLD (Anruf wurde auf Halten gesetzt) LINECALLSTATE_CONFERENCED (Anruf in ein Konferenzgespräch geschaltet) LINECALLSTATE_ONHOLDPENDCONF (Anruf vor Konferenzschaltung auf Halten gesetzt) LINECALLSTATE_ONHOLDPENDTRANSFER (Anruf vor Vermittlung auf Halten gesetzt) LINECALLSTATE_DISCONNECTED (Verbindung wurde vom anderen Gesprächsteilnehmer getrennt) LINECALLSTATE_UNKNOWN (Anrufstatus unbekannt)
dwCallStateMode	immer 0
dwCallPrivilege	die Anwendungsberechtigung für diesen Anruf
dwCallFeatures	Die für den durch **dwCallState** angegebenen Anrufstatus verfügbaren Anruffunktionen. TAPI gibt alle möglichen Funktionen an, es können jedoch nur die in der Struktur LINEADDRESSCAPS unter **dwCallFeatures** aufgeführten Funktionen verwendet werden.
dwDevSpecificSize	0
dwDevSpecificOffset	0
dwCallFeatures2	0
tStateEntryTime	Nullen

LINEDEVCAPS

Diese Struktur wird von der Funktion **lineGetDevCaps** zurückgegeben. In der folgenden Tabelle sind die Werte aufgeführt, die für Leitungen in Verbindung mit dem IP Office-TAPI-Treiber zurückgegeben werden.

- **Hinweis:** Die Tabelle enthält nicht alle Details dieser Struktur. Eine ausführliche Erläuterung der möglichen Werte für LINEDEVCAPS finden Sie in der Dokumentation zu Microsoft TAPI.

Element	Beschreibung/Wert
dwProviderInfoSize	Name des Providers, d.h. Name des TSP
dwSwitchInfoSize	0
dwPermanentLineID	Eindeutige, von Windows zugewiesene ID
dwLineNameSize	Leitungsname
dwStringFormat	Dient zur Rückgabe des Zeichenfolgenformats: STRINGFORMAT_ASCII
dwAddressModes	Dient zur Rückgabe des Adressmodus: LINEADDRESSMODE_ADDRESSID
dwNumAddresses	1
dwBearerModes	Dient zur Rückgabe der möglichen Dienstmodi: LINEBEARERMODE_VOICE LINEBEARERMODE_SPEECH
dwMaxRate	0
dwMediaModes	Dient zur Rückgabe des Medienmodus: LINEMEDIAMODE_INTERACTIVEVOICE
dwGenerateToneModes	Dient zur Rückgabe des Ton-Generierungs-Modus: LINETONEMODE_BEEP
dwGenerateToneMaxNumFreq	0
dwMonitorToneMaxNumFreq	1

dwMonitorToneMaxNumEntries	1
dwGatherDigitsMinTimeout	0
dwGatherDigitsMaxTimeout	0
dwMedCtlDigitMaxListSize	0
dwMedCtlMediaMaxListSize	0
dwMedCtlToneMaxListSize	0
dwMedCtlCallStateMaxListSize	0
dwDevCapFlags	Dient zur Rückgabe der Markierungen für die Gerätewerte: LINEDEVCAPFLAGS_CLOSEDROP LINEDEVCAPFLAGS_DIALBILLING LINEDEVCAPFLAGS_DIALQUIET LINEDEVCAPFLAGS_DIALDUALTONE
dwMaxNumActiveCalls	9
dwAnswerMode	Dient zur Rückgabe des Antwortmodus: LINEANSWERMODE_NONE
dwRingModes	1
dwLineStates	Dient zur Rückgabe des Leitungsstatus: LINEDEVSTATE_RINGING LINEDEVSTATE_CONNECTED LINEDEVSTATE_DISCONNECTED LINEDEVSTATE_INSERVICE LINEDEVSTATE_OUTOFSERVICE LINEDEVSTATE_OPEN LINEDEVSTATE_CLOSE LINEDEVSTATE_REINIT LINEDEVSTATE_TRANSLATECHNGE LINEDEVSTATE_REMOVED
dwUUIAcceptSize	0
dwUUIAnswerSize	100
dwUUIMakeCallSize	100

dwUUIDropSize	100
dwUUISendUserUserInfoSize	100
dwUUICallInfoSize	Anrufdatengröße Benutzer zu Benutzer: 100
dwNumTerminals	0
dwTerminalCapsSize	0
dwTerminalCapsOffset	0
dwTerminalTextEntrySize	0
dwTerminalTextSize	0
dwTerminalTextOffset	0
dwDevSpecificSize	0
dwDevSpecificOffset	0
dwLineFeatures	Dient zur Rückgabe der Leitungsfunktion: LINEFEATURE_MAKECALL
dwSettableDevStatus	0
dwDeviceClassesSize	tapi\line
PermanentLineGuide	Relevant ab TAPI Version 2.2.
dwAddressTypes	Relevant ab TAPI Version 3.0.
ProtocolGuide	Relevant ab TAPI Version 3.0.
dwAvailableTracking	Relevant ab TAPI Version 3.0.

TAPI-Ereignisse (Meldungen)

LINE_APPNEWCALL

Ein neuer Anruf wurde erstellt.

LINE_CALLINFO

Informationen in der Struktur LINECALLINFO haben sich geändert.

LINE_CALLSTATE

Der Status des Anrufs hat sich geändert. Eine Liste der unterstützten Status finden Sie in der Struktur LINEADDRESSCAPS unter **dwCallStates**.

LINE_LINEDEVSTATE

Der Status der Leitung hat sich geändert. Für den zweiten Parameter bestehen folgende Möglichkeiten:

- LINEDEVSTATE_DEVSPECIFIC: Gerätespezifische Informationen haben sich geändert.
- LINEDEVSTATE_CONNECTED, LINEDEVSTATE_DISCONNECTED: Der Verbindungsstatus der Leitung hat sich geändert.
- LINEDEVSTATE_OUTOFSERVICE: Die Kommunikation zwischen TSP und Vermittlung wurde unterbrochen. Diese Leitung ist nun außer Betrieb.
- LINEDEVSTATE_INSERVICE: Die Kommunikation zwischen TSP und Vermittlung wurde unterbrochen, die Verbindung konnte jedoch wiederhergestellt werden, und die Leitung ist wieder in Betrieb.
- LINEDEVSTATE_RINGING: Die Vermittlung hat festgestellt, dass das Telefon des angerufenen Teilnehmers klingelt.

LINE_DEVSPECIFIC

Informiert die Anwendung über gerätespezifische Ereignisse in einer Leitung, Adresse oder einem Anruf. Die Anwendung wird durch diese Meldung aufgefordert, die Funktion **lineGetLineDevStatus** aufzurufen und den gerätespezifischen Puffer auf Änderungen zu überprüfen.

LINE_ADDRESSSTATE

Der Status einer Adresse wurde in einer Leitung geändert, die zurzeit von der Anwendung geöffnet ist.

TAPI 3.0

TAPI

Das TAPI-Objekt wird von **CoCreateInstance** erstellt. Alle anderen Objekte in TAPI 3.0 werden von TAPI 3.0 erstellt.

ITTAPI

Die Schnittstelle **ITTAPI** dient als Basisschnittstelle für das TAPI-Objekt.

Initialize

Dies ist die erste TAPI-Funktion, die zur Initialisierung von TAPI aufgerufen werden sollte.

```
HRESULT
Initialize();
```

Shutdown

Hiermit wird eine TAPI-Sitzung geschlossen. In der Regel wird diese Funktion beim Herunterfahren Ihrer Anwendung aufgerufen.

```
HRESULT
Shutdown();
```

EnumerateAddresses

Diese Methode listet die aktuell verfügbaren Adressen auf.

```
HRESULT
EnumerateAddresses ( IEnumAddress **ppEnumAddress );
```

RegisterCallNotifications

Hiermit wird festgelegt, welche neuen Anrufbenachrichtigungen eine Anwendung empfängt. Die Anwendung muss diese Methode für jede Adresse aufrufen und die unterstützten Medientypen sowie die angeforderten Berechtigungen müssen angegeben werden.

```
HRESULT RegisterCallNotifications(
ITAddress *pAddress,
VARIANT_BOOLfMonitor,
VARIANT_BOOLfOwner,
long lMediaTypes,
long lCallbackInstance,
long *plRegister
);
```

put_EventFilter

Die Methode **put_EventFilter** stellt die Ereignis-Filtermaske ein.

```
HRESULT
put_EventFilter ( long lFilterMask);
```

Address

Das Adressobjekt (**Address**) stellt eine Einheit dar, die Anrufe senden und empfangen kann.

ITAddress

Diese Schnittstelle dient als Basisschnittstelle für das Objekt **Address**.

get_AddressName

Ruft den Anzeigename der Adresse ab.

```
HRESULT
get_AddressName (BSTR *ppName );
```

get_DialableAddress

Die Methode **get_DialableAddress** ruft den Wert **BSTR** ab, der zur Herstellung einer Verbindung mit dieser Adresse verwendet werden kann.

```
HRESULT
get_DialableAddress (
BSTR *pDialableAddress
);
```

get_ServiceProviderName

Die Methode **get_ServiceProviderName** ruft den Namen des Telefonie-Dienstanbieters (Telephony Service Provider, TSP) ab, der diese Adresse unterstützt. Beispiel: **Unimdm.tsp** für den Dienstanbieter Unimodem oder **H323.tsp** für den Dienstanbieter H323.

```
HRESULT
get_ServiceProviderName (
BSTR *ppName
);
```

CreateCall

Die Methode **CreateCall** erstellt ein neues Call-Objekt, das für einen ausgehenden Anruf verwendet werden kann und einen Zeiger auf die Objektschnittstelle **ITBasicCallControl** zurückgibt.

```
HRESULT
CreateCall (
BSTR *pDialableAddress,
Long lAddressType,
Long lMediaTypes,
ITBasicCallControl**ppCall
);
```

IEnumAddress

Stellt Aufzählungsmethoden nach COM-Standard für die **ITAddress**-Schnittstelle bereit.

Next

Die Methode **Next** ruft die nächste angegebene Anzahl an Elementen in der Aufzählungsreihenfolge ab.

```
HRESULT
Next(
ULONG celt,
ITAddress**ppElements,
ULONG *pceltFetched
);
```

ITMediaSupport

Die Schnittstelle **ITMediaSupport** stellt Methoden bereit, mit denen eine Anwendung ermitteln kann, welche Medien das zu dieser Schnittstelle gehörende Address-Objekt unterstützt.

get_MediaTypes

Die Methode **get_MediaTypes** ruft die an der aktuellen Adresse unterstützten Medientypen ab.

```
HRESULT
get_MediaTypes (
long *plMediaTypes
);
```

Terminal

Das Terminal-Objekt stellt die Quelle eines mit einem Anruf oder einer Kommunikationssitzung verbundenen Medienstroms dar.

Call

Das Anrufobjekt (**Call**) stellt die Adressenverbindung zwischen der lokalen Adresse und einer oder mehreren anderen Adressen dar.

ITCallInfo

Die Schnittstelle **ITCallInfo** dient zum Abrufen und Festlegen verschiedener Informationen im Zusammenhang mit einem Call-Objekt.

get_Address

Die Methode **get_Address** ruft einen Zeiger auf die Schnittstelle **ITAddress** des Address-Objekts ab.

```
HRESULT
get_Address (
ITAddress **ppAddress
);
```

get_CallState

Die Methode **get_CallState** ruft einen Zeiger auf den aktuellen Anrufstatus ab, wie beispielsweise CS_IDLE.

```
HRESULT
get_CallState (
CALL_STATE *pCallState
);
```

get_CallInfoString

Die Methode **get_CallInfoString** ruft Anrufinformationen in Form einer Zeichenfolge ab, wie beispielsweise den Anzeigenamen der Adresse.

```
HRESULT
get_CallInfoString(
CALLINFO_STRING CallInfoString,
BSTR *ppCallInfoString
);
```

SetCallInfoBuffer

TAPI 3.0 (Windows 2000) unterstützt diese Funktion nur für inaktive Anrufe mit dem Status IDLE. Dieses Problem wurde in TAPI 3.1 (Windows XP) gelöst. In dieser Version können Anrufdaten für verbundene Anrufe festgelegt werden, indem CIB_CALLDATABUFFER als Parameter **CallInfoBuffer** übergeben wird.

```
HRESULT
SetCallInfoBuffer (
CALLINFO_BUFFERCallInfoBuffer,
DWORD dwSize
BYTE* pCallInfoBuffer
);
```

ITBasicCallControl

Die Schnittstelle **ITBasicCallControl** ermöglicht der Anwendung das Herstellen einer Verbindung zu einem Call-Objekt sowie das Senden von Antworten und die Ausführung grundlegender Telefoniefunktionen.

Connect

Die Methode **Connect** versucht, die Verbindung eines ausgehenden Anrufs abzuschließen.

```
HRESULT
Connect(
VARIANT_BOOLfSync
);
```

Answer

Die Methode **Answer** beantwortet eingehende Anrufe. Diese Methode kann nur erfolgreich ausgeführt werden, wenn der Anrufstatus CS_OFFERING lautet.

```
HRESULT
Answer();
```

Disconnect

Die Methode **Disconnect** trennt die Anrufverbindung. Nach erfolgreichem Abschluss der Methode ändert sich der Anrufstatus in CS_DISCONNECTED.

```
HRESULT
Disconnect(
DISCONNECT_CODE code
);
```

Hold

Die Methode **Hold** kann einen Anruf halten und einen gehaltenen Anruf aktivieren.

```
HRESULT
Hold(
VARIANT_BOOLfHold
);
```

SwapHold

Die Methode **SwapHold** tauscht den aktiven Anruf gegen den angegebenen gehaltenen Anruf.

```
HRESULT
SwapHold(
ITBasicCallControl *pCall
);
```

ParkDirect

Die Methode **ParkDirect** parkt den Anruf an einer angegebenen Adresse.

```
HRESULT
ParkDirect(
BSTR pParkAddress
);
```

Unpark

Die Methode **Unpark** holt einen Anruf aus dem geparkten Status zurück.

```
HRESULT
Unpark();
```

BlindTransfer

Die Methode **BlindTransfer** führt eine nicht überwachte Vermittlung des angegebenen Anrufs an die angegebene Adresse aus.

```
HRESULT BlindTransfer(
BSTR pDestAddress
);
```

Transfer

Die Methode **Transfer** vermittelt den aktuellen Anruf an die Zieladresse.

```
HRESULT Transfer(
ITBasicCallControl*pCall,
VARIANT_BOOLfSync
);
```

Finish

Die Methode **Finish** wird bei einem Rückfrageruf zum Fertigstellen eines Konferenzgesprächs oder einer Vermittlung aufgerufen.

```
HRESULT Finish(
FINISH_MODE finishMode
);
```

Conference

Die Methode **Conference** schaltet einen Rückfrageruf für das Konferenzgespräch, an dem der aktuelle Anruf teilnimmt.

```
HRESULT Conference(
ITBasicCallControl *pCall,
VARIANT_BOOL fSync
);
```

RemoveFromConference

Die Methode **RemoveFromConference** entfernt den Anruf ggf. aus einem Konferenzgespräch.

```
HRESULT RemoveFromConference();
```

ITCallStateEvent

Die Schnittstelle **ITCallStateEvent** umfasst Methoden zum Abrufen der Beschreibung von Anrufstatusereignissen.

get_Cause

Die Methode **get_Cause** ruft die mit diesem Ereignis verbundene Ursache ab.

```
HRESULT
get_Cause (
CALL_STATE_EVENT_CAUSE *pCEC
);
```

get_State

Die Methode **get_State** ruft Informationen zum neuen Anrufstatus ab.

```
HRESULT
get_State (
CALL_STATE *pCallState
);
```

get_Call

Die Methode **get_Call** ruft einen Zeiger auf die Anrufinformationsschnittstelle ab, an der das Ereignis eingetreten ist.

```
HRESULT
get_Call
ITCallInfo**ppCallInfo
);
```

ITCallNotificationEvent

Die Schnittstelle **ITCallNotificationEvent** umfasst Methoden zum Abrufen der Beschreibung von Anrufereignissen.

get_Call

Die Methode **get_Call** gibt die **ITCallInfo**-Schnittstelle zurück, an der das Anrufereignis eingetreten ist.

```
HRESULT
get_Call
ITCallInfo**ppCall
);
```

ITCallInfoChangeEvent

Die Schnittstelle **ITCallInfoChangeEvent** umfasst Methoden zum Abrufen der Beschreibung von Ereignissen, wenn sich Anrufinformationen ändern.

get_Call

Die Methode **get_Call** gibt die **ITCallInfo**-Schnittstelle zurück, an der sich Anrufinformationen geändert haben.

```
HRESULT
get_Call
ITCallInfo**ppCall
);
```

CallHub

Das Call Hub-Objekt (CallHub) stellt Methoden zum Abrufen von Informationen zu Teilnehmern an einem Konferenzgespräch bereit. Call Hubs werden von IP Office nicht unterstützt. Call Hub-Ereignisse können empfangen werden, sollten aber ignoriert werden.

TAPI 3-Aufzählungstypen

CALL_STATE

CALL_STATE wird von den Methoden **ITCallInfo::get_CallState** und **ITCallStateEvent::get_State** verwendet.

Element	Wert	Beschreibung
CS_IDLE	0	Der Anruf wurde erstellt, **Connect** wurde jedoch noch nicht aufgerufen. Ein Anruf kann nie in diesen Status übergehen.
CS_INPROGRESS	1	**Connect** wurde aufgerufen und der Dienstanbieter versucht, eine Verbindung herzustellen. Dieser Status ist nur für ausgehende Anrufe gültig. Diese Meldung ist optional, da möglicherweise der direkte Übergang zum verbundenen Status erfolgt.
CS_CONNECTED	2	Der Anruf wurde verbunden und das Gespräch kann begonnen werden.
CS_DISCONNECTED	3	Die Verbindung wurde getrennt. Es gibt verschiedene Ursachen für die Trennung einer Verbindung. Siehe folgende Tabelle der gültigen Anrufstatusübergänge.
CS_OFFERING	4	Ein neuer Anruf liegt vor und wartet an einer Anwendung. Wenn die Anwendung über Besitzerrechte für den Anruf verfügt, kann sie entweder **Answer** oder **Disconnect** aufrufen, während der Anruf sich im wartenden Status befindet.
CS_HOLD	5	Der Anruf wird gehalten.
CS_QUEUED	6	Der Anruf wird in die Warteschlange gestellt.

CALLINFO_STRING

CALLINFO_STRING wird von den **ITCallInfo**-Methoden verwendet, die zum Abrufen und Festlegen von Anrufinformationen mit Hilfe von Zeichenfolgen dienen.

Element	Wert	Beschreibung
CIS_CALLERIDNAME	0	Der Name des Anrufers.
CIS_CALLERIDNUMBER	1	Die Nummer des Anrufers.
CIS_CALLEDIDNAME	2	Der Name des angerufenen Standorts.
CIS_CALLEDIDNUMBER	3	Die Nummer des angerufenen Standorts.
CIS_CONNECTEDIDNAME	4	Der Name des verbundenen Standorts.
CIS_CONNECTEDIDNUMBER	5	Die Nummer des verbundenen Standorts.
CIS_REDIRECTIONIDNAME	6	Der Name des Standorts, an den der Anruf umgeleitet wurde.
CIS_REDIRECTIONIDNUMBER	7	Die Nummer des Standorts, an den der Anruf umgeleitet wurde.
CIS_REDIRECTINGIDNAME	8	Der Name des Standorts, der den Anruf umgeleitet hat.
CIS_REDIRECTINGIDNUMBER	9	Die Nummer des Standorts, der den Anruf umgeleitet hat.
CIS_CALLEDPARTYFRIENDLYNAME	10	Der angezeigte Name der angerufenen Adresse.
CIS_COMMENT	11	Ein Kommentar zu dem Anruf von der Anwendung, von der der Anruf ausging.
CIS_DISPLAYABLEADDRESS	12	Die Anzeigeversion der angerufenen Adresse bzw. der Adresse des Anrufers.
CIS_CALLINGPARTYID	13	Die ID des Anrufers.

DISCONNECT_CODE

DISCONNECT_CODE wird von der Methode **ITBasicCallControl::Disconnect** verwendet.

Element	Wert	Beschreibung
DC_NORMAL	0	Die Verbindung wird unter normalen Bedingungen getrennt, d.h. das Gespräch wurde beendet.
DC_NOANSWER	1	Die Verbindung wird getrennt, da der Anruf nicht beantwortet wurde. (In einer Anwendung kann beispielsweise eingestellt werden, wie lange auf eine Antwort gewartet wird. Wenn keine Antwort erfolgt, ruft die Anwendung **Disconnect** mit dem Code NOANSWER auf.)
DC_REJECTED	2	Der wartende Anruf wurde abgelehnt.

CALL_STATE_EVENT_CAUSE

CALL_STATE_EVENT_CAUSE wird von der Methode **ITCallStateEvent::get_Cause** zurückgegeben.

Element	Wert	Beschreibung
CEC_NONE	0	Es ist kein Anrufereignis aufgetreten.
CEC_DISCONNECT_NORMAL	1	Die Verbindung wurde unter normalen Bedingungen getrennt (d.h. nach Beendung des Gesprächs).
CEC_DISCONNECT_BUSY	2	Ein ausgehender Anruf konnte nicht durchgestellt werden, da die angerufene Nummer besetzt war.
CEC_DISCONNECT_BADADDRESS	3	Ein ausgehender Anruf ist fehlgeschlagen, da die Zieladresse fehlerhaft war.
CEC_DISCONNECT_NOANSWER	4	Ein ausgehender Anruf ist fehlgeschlagen, da die angerufene Nummer nicht geantwortet hat.
CEC_DISCONNECT_CANCELLED	5	Ein ausgehender Anruf ist fehlgeschlagen, da der Anrufer die Verbindung getrennt hat.
CEC_DISCONNECT_REJECTED	6	Der ausgehende Anruf wurde vom angerufenen Teilnehmer nicht angenommen.
CEC_DISCONNECT_FAILED	7	Die Verbindung konnte aufgrund anderer Ursachen nicht hergestellt werden.

IP Office Media Service Provider

Info zum MSP

Der IP Office Media Service Provider ist doppelt nützlich. Er bietet Media-Streaming-Funktionen, mit denen TAPI 3-Anwendungen Sprachdaten für Anrufe in spezifischen Typen von Benutzerleitungen senden und empfangen können. Außerdem ermöglicht er Anwendungen den Zugriff auf gerätespezifische Funktionen von IP Office.

Verwenden des MSP

Die Schnittstellen des Media Service Provider sind in den MSDN-Bibliotheken dokumentiert. Das Beispiel **DevSpice** auf der SDK-CD gibt einen Einblick in die Verwendung des MSP für Media-Streaming- und gerätespezifische Funktionen. Der MSP ist für jede TAPI-Adresse verfügbar, die in Ihrer TAPI 3-Anwendung angezeigt wird. Die Media-Streaming-Funktionen stehen nur für Adressen zur Verfügung, die als Wave-Benutzer gekennzeichnet sind. Die Namen von WAVE-Benutzern beginnen mit "TAPI:" (z.B. "TAPI:201"). Sie können beliebig viele WAVE-Benutzer erstellen. Für jeden Wave-Benutzer ist jedoch eine Wave-Treiber-Lizenz erforderlich, um das Senden von Medienströmen an diesen Benutzer zu ermöglichen.

Verwenden der gerätespezifischen Schnittstellen

Die gerätespezifischen Schnittstellen werden im Adressobjekt (**Address**) und im Anrufobjekt (**Call**) des MSP implementiert. In TAPI 3.0 werden Abfragen für Schnittstellen, die nicht erkannt werden, an den MSP übergeben. Wenn Sie beispielsweise über einen Zeiger auf eine Schnittstelle **ITAddress** verfügen, können Sie mit Hilfe von **QueryInterface** einen Zeiger auf die Schnittstelle **ITDivert** abrufen. Siehe folgenden Code aus dem Beispiel **DevSpice**:

```
ITDivert* pDivert = NULL;

if( SUCCEEDED( gpAddress->QueryInterface( IID_ITDIVERT,

(void**)&pDivert)))

{

DWORD dwDivertSettings = 0;
if( FAILED( pDivert-> GetDivertSettings(
&dwDivertSettings)))

{
```

Folgende Schnittstellen sind vom Adressobjekt verfügbar:

- ITACDAgent
- ITDivert
- ITGroup

Folgende Schnittstelle ist vom Anrufobjekt verfügbar:

- ITPlay

Des Weiteren fungiert das Adressobjekt als Verbindungspunkt-Container für private Ereignisse in IP Office. Die Verbindungspunktschnittstelle mit dem Namen **IPOfficePrivateEvent** ist in der Datei **interfaces.h** des Beispiels **DevSpice** verfügbar. Nähere Informationen zu diesen Schnittstellen finden Sie im Folgenden.

ITACDAgent

IsLoggedIn(void)	Gibt S_TRUE zurück, wenn der Benutzer angemeldet ist, und S_FALSE, wenn der Benutzer abgemeldet ist.
LogOut(void)	Meldet den Benutzer bei dieser Leitung ab. Für den Benutzer muss in Manager die Einstellung **Login erzwingen** aktiviert sein.
Login(BSTR extn)	Meldet den Benutzer an der gegebenen Nebenstelle an.
CallListen(BSTR extn)	Hört den Anruf an der gegebenen Nebenstelle ab. Für den Benutzer muss in Manager die Einstellung **Aufschalten erlaubt** aktiviert sein.
Intrude(BSTR extn)	Schaltet den aktuellen Benutzer in den Anruf an der gegebenen Nebenstelle. Für den Benutzer muss in Manager die Einstellung **Aufschalten erlaubt** aktiviert sein.
SetAccountCode(BSTR extn)	Legt den Verrechnungscode für den aktuellen Anruf fest.

ITGroup

Diese Schnittstelle bietet Funktionen zur Aktivierung bzw. Deaktivierung von Gruppenmitgliedschaften eines Benutzers sowie zum Abfangen von Anrufen an anderen Telefongeräten der Gruppe.

PickupAny(void)	Entspricht der Ausführung des Funktionscodes **CallPickupAny** am Benutzerendgerät. Nähere Informationen finden Sie unter *Manager*.
PickupGroup(void)	Entspricht der Ausführung des Funktionscodes **CallPickupGroup** am Benutzerendgerät.
PickupExtn(BSTR extn)	Entspricht der Ausführung des Funktionscodes **CallPickupExtn** am Benutzerendgerät.
PickupMembers(BSTR extn)	Entspricht der Ausführung des Funktionscodes **CallPickupMembers** am Benutzerendgerät.
Enable(BSTR groupextn)	Aktiviert die Mitgliedschaft des Benutzers bei der gegebenen Gruppe. Wenn **groupextn** eine leere Zeichenfolge enthält, werden alle vorhandenen Gruppenmitgliedschaften des Benutzers aktiviert.
Disable(BSTR groupextn)	Deaktiviert die Mitgliedschaft des Benutzers bei der gegebenen Gruppe. Wenn **groupextn** eine leere Zeichenfolge enthält, werden alle vorhandenen Gruppenmitgliedschaften des Benutzers deaktiviert.

ITDivert

Diese Schnittstelle bietet Funktionen zum Abrufen und Festlegen der Weiterleitungseinstellungen für die Adresse.

GetDivertAllDestination(BSTR* pDestination)	Ruf das aktuelle Ziel für **Alle weiterleiten** ab und gibt das Ergebnis im Wert **pDestination** zurück.
SetDivertAllDestination(BSTR dest)	
GetDivertSettings(DWORD* pdwDivertSets)	Diese Funktion legt Bit für den Wert DWORD fest, auf den **pdwDivertSets** verweist. So wird gekennzeichnet, welche Weiterleitungseinstellungen zurzeit aktiv sind. Die Bit werden mit Hilfe des weiter unten beschriebenen Enumerators IP_OFFICE_DIVERT_SETTINGS definiert.
SetForwardAll(VARIANT_BOOL bOn)	Aktiviert/deaktiviert die Einstellung **Alle weiterleiten** für diesen Benutzer.
SetForwardBusy(VARIANT_BOOL bOn)	Aktiviert/deaktiviert die Einstellung **Rufweiterleitung bei Besetzt** für diesen Benutzer.
SetForwardNoAnswer(VARIANT_BOOL bOn)	Aktiviert/deaktiviert die Einstellung **Rufweiterleitung nach Zeit** für diesen Benutzer.
SetDoNotDisturb(VARIANT_BOOL bOn)	Aktiviert/deaktiviert die Einstellung **Nicht stören** für diesen Benutzer.

Der Enumerator IP_OFFICE_DIVERT_SETTINGS wird wie folgt definiert:

```
typedef enum
{
IPOFF_FWDALL = 0x01,
IPOFF_FWDBUSY = 0x02,
IPOFF_NOANSWER = 0x04,
IPOFF_DND = 0x08,
IPOFF_DESTINATION = 0x10
} IP_OFFICE_DIVERT_SETTINGS;
```

Das Ergebnis 14 (0xe) von **GetDivertSettings** bedeutet somit, dass für den Benutzer die Einstellungen **Rufweiterleitung bei Besetzt**, **Rufweiterleitung nach Zeit** und **Nicht stören** aktiviert sind. Der Wert IPOFF_DESTINATION wird nicht von **GetDivertSettings** verwendet, sondern lediglich von der Funktion **Fire_DivertSettingsChanged** an der Schnittstelle **IPOfficePrivateEvents**.

ITPlay

Die Schnittstelle **ITPlay** ist im Objekt **MSP Call** implementiert. Sie ermöglicht das Aufzeichnen und Abspielen von Wave-Dateien.

StartPlay(BSTR FileName)	**FileName** muss den vollständigen Pfad der Wave-Datei wiedergeben, die abgespielt werden soll.
StopPlay()	
StartRecord(BSTR FileName)	**FileName** muss den vollständigen Pfad der Wave-Datei für die Aufzeichnung wiedergeben.
StopRecord()	

Das Abspielen und Aufzeichnen kann zu einem beliebigen Zeitpunkt des Anrufs gestoppt und gestartet werden. Für die Aufzeichnung wird jedoch nur eine Datei pro Anruf verwendet. Wenn die Aufzeichnung gestoppt und erneut gestartet wird, werden die Daten an diese Datei angehängt. Abspiel- und Aufzeichnungsvorgänge sollten nicht gleichzeitig ausgeführt werden. Wenn dies erforderlich ist, können Sie Endgeräte in den Anruf schalten, über die Audiodaten aus einer Datei bereitgestellt bzw. Autiodaten in einer Datei aufgezeichnet werden. In TAPI 3.1 wird dieser Vorgang mit Hilfe von Datei-Streaming-Endgeräten vereinfacht.

IPOfficePrivateEvents

Hierbei handelt es sich um eine Verbindungspunktschnittstelle, die vom MSP für Ereignisberichte verwendet wird. Informationen zur Registrierung für und zur Handhabung von privaten Ereignissen finden Sie im Beispiel *DevSpice*.

OnUserLogin(void)	Ausgelöst, wenn ein Agent (ein Benutzer, für den in Manager die Einstellung **Login erzwingen** aktiviert ist) sich anmeldet.
OnUserLogout(void)	Ausgelöst, wenn ein Agent sich abmeldet.
OnDivertSettingsChanged(DWORD dwDivertSettings)	Ausgelöst, wenn der Benutzer eine seiner Umleitungseinstellungen (wie **Nicht stören** oder **Rufweiterleitung bei Besetzt**) oder das Ziel für **Alle weiterleiten** ändert. Die Bit in der Variablen **dwDivertSettings** werden unter Verwendung von IP_OFFICE_DIVERT_SETTINGS (oben beschrieben) festgelegt.
OnGroupChanged(DWORD dwGroupCount)	Ausgelöst, wenn der Benutzer eine Gruppenmitgliedschaft aktiviert oder deaktiviert. Mit dem Wert **dwGroupCount** wird die Anzahl der Gruppen angegeben, für die der Benutzer als Mitglied aktiviert ist.
OnVoiceMail(DWORD dwNumMessages)	Ausgelöst, wenn sich die Anzahl der neuen VoiceMail-Nachrichten für einen Benutzer ändert. Der neue Wert wird mit dem Parameter **dwNumMessages** ausgegeben.

Verwenden der Media-Streaming-Funktionen des MSP

Der IP Office MSP ermöglicht die Handhabung von Media-Streaming an jeden beliebigen Wave-Benutzer. Hierzu müssen Sie Endgeräte für die Datenströme schalten, die der MSP für einen Anruf bereitstellt. Nähere Informationen zur Vorgehensweise finden Sie im MSDN (Microsoft Developer Network) und ein Beispiel im Beispiel **DevSpice**. Beachten Sie, dass IP Office-Datenströme bidirektional sind und nur ein Datenstrom pro Anruf vorhanden ist. Dies bedeutet, dass jeweils ein Sende- und ein Empfangsgerät für denselben Datenstrom akzeptiert werden.

Der MSP beinhaltet die Funktionen des IP Office-Wave-Treibers. Der IP Office-Wave-Treiber muss auf jedem Gerät installiert sein, mit dem Medienströme an Benutzer gesendet werden sollen. Wenn Sie kein Media-Streaming durchführen, sondern lediglich Wave-Benutzer mit TAPI überwachen möchten, stellen Sie sicher, dass der Wave-Treiber nicht installiert ist. Andernfalls nutzen Sie für jede geöffnete Wave-Benutzerleitung unnötig eine Wave-Lizenz.

Index

0

0.n 21
0x00020000 24
0x01 61
0x02 61
0x04 61
0x08 61
0x10 61
0xe 61
0xFFFFFFFF 37

2

2.x 5

A

Abhören 16
Abhören/Aufschalten 6
Abmelden 60
Abrufen 19
  LINECALLSTATUS 19
ACD 14
  Protokoll 14
ACD-Warteschlangen 7
ACD-Warteschlangen (Kontrollkästchen) 7
  Überprüfen 7
Address 9, 46, 48, 59
Adressobjekt 47
Alles umleiten 61, 62
Analog 26
  LINEMEDIAMODE_UNKNOWN 26
Analyse 25, 43
  devspecific 43
Anbietername 40
  Anzeige 40
Änderungen 6
Angeben 7
  TAPI 7
Angerufene 43
Anmelden 14, 21, 60, 62
  ACD 14
  Aus 14
Anmeldeprotokoll 14
Anrufer-ID 37
Anrufinformationen 27
AnrufMithören 60
Answer 49, 55
Anwendungsprogrammierschnittstellen 5
anwendungsspezifisch 27
  einrichten 27
Anzeige 9, 40
  Anbietername 40
  Leitungsname 40
  TAPI-Anwendung 9
API 5
AS_OUTOFSERVICE 9
Aufhebung der Zuweisung 13
Aufschalten 16
Aufschalten erlaubt 60
Aufzählungstypen 5
Aus 14
  Anmelden 14
Ausführung 60
  Heranholen beliebig (Funktionscode) 60
  Heranholen Nst (Funktionscode) 60
  Heranholen Sammelanschluss beliebig (Funktionscode) 60
Ausgabe 40, 53
  dev 40
  ITCallInfo 53
Auswählen 7
  Drittanwender 7
  Einzelbenutzer 7
Avaya 5

B

Benutzer 8, 40
Beschreibung/Wert 30, 36, 37, 38, 39, 40
Betreff 48
  Call 48
beziehen 30, 40
  IP Office TAPI 30, 40
Bezug auf 5, 7
  CTI Link-Installationshandbuch 7
  IP Office CTI Link-Installationshandbuch 5
Bidirektional 63
BlindTransfer 49
BSTR 46, 48, 61
  Abrufen 46
BSTR dest 61
BSTR FileName 62
BSTR groupextn 60
BSTR pDestAddress 49
BSTR pParkAddress 49
BSTR-Nst. 60
Buf 14, 15, 16
BYTE 48

C

Call 9, 14, 24, 45, 46, 48, 59
  Betreff 48
  Initialisierung von TAPI 24, 45
  lineDevSpecific 14
  TAPI 9
Call lineCompleteTransfer 28
Call lineDial 28
Call lineMonitorDigits 25
Call lineMonitorTone 25
CALL_STATE 48, 52, 55
CALL_STATE enum 55
CALL_STATE_EVENT_CAUSE 52, 57
CALL_STATE_EVENT_CAUSE enum 57
CallHub 53
CallHub-Ereignisse 53
CALLINFO_BUFFER CallInfoBuffer 48
CALLINFO_STRING 56
CALLINFO_STRING CallInfoString 48
CALLINFO_STRING enum 56
CallInfoBuffer 48
CEC_DISCONNECT_BADADDRESS 57
CEC_DISCONNECT_BUSY 57
CEC_DISCONNECT_CANCELLED 57
CEC_DISCONNECT_FAILED 57
CEC_DISCONNECT_NOANSWER 57
CEC_DISCONNECT_NORMAL 57
CEC_DISCONNECT_REJECTED 57

CEC_NONE 57
CIB_CALLDATABUFFER 48
  Weitergeben 48
CIS_CALLEDIDNAME 56
CIS_CALLEDIDNUMBER 56
CIS_CALLEDPARTYFRIENDLYNAME 56
CIS_CALLERIDNAME 56
CIS_CALLERIDNUMBER 56
CIS_CALLINGPARTYID 56
CIS_COMMENT 56
CIS_CONNECTEDIDNAME 56
CIS_CONNECTEDIDNUMBER 56
CIS_DISPLAYABLEADDRESS 56
CIS_REDIRECTINGIDNAME 56
CIS_REDIRECTINGIDNUMBER 56
CIS_REDIRECTIONIDNAME 56
CIS_REDIRECTIONIDNUMBER 56
CoCreateInstance 45
COM-Standard 47
Connect 49, 55
Const 15
CreateCall 46
CS_CONNECTED 55
CS_DISCONNECTED 49, 55
CS_HOLD 55
CS_IDLE 48, 55
CS_INPROGRESS 55
CS_OFFERING 49, 55
CS_QUEUED 55
CTI Link Pro 5, 7
CTI Link-Installationshandbuch 5, 7
  Bezug auf 7
CTI TAPI Linkpro-Lizenz 7
  Installieren 7

D

D 15, 16
Dateiname 62
DC_NOANSWER 57
DC_NORMAL 57
DC_REJECTED 57
Deaktivieren 16
Dev 40
  Ausgabe 40
Devspecific 6, 14, 15, 21, 43
  Analyse 43
  Puffer 14
DevSpice 59, 62, 63
  interfaces.h (Datei) 59
  Siehe 62
Disconnect 49, 55, 57
DISCONNECT_CODE 49, 57
DISCONNECT_CODE enum 57
Dokument 5
Drittanwender 7
  Auswählen 7
Drittanwendermodus 7
DwAddrCapFlags 30
DwAddressFeatures 30, 36
DwAddressID 18, 29, 37
DwAddressModes 40
DwAddressSharing 30
DwAddressStates 28, 30
DwAddressTypes 40
DwAnswerMode 40
DwAppSpecific 37
DwAvailableMediaModes 30
DwAvailableTracking 40
DwBearerMode 37
DwBearerModes 40
DwBusyModes 30
DwCallCompletionConds 30
DwCallCompletionModes 30
DwCallDataOffset 37
DwCalledIDAddressType 37
DwCalledIDFlags 30
DwCalledPartyOffset 38
DwCallerIDAddressType 37
DwCallerIDFlags 30
DwCallFeatures 30, 39
DwCallFeatures2 30, 39
DwCallID 37
DwCallInfoStates 30
DwCallingPartyIDOffset 38
DwCallParamFlags 37, 38
DwCallPrivilege 39
DwCallState 39
DwCallStateMode 39
DwCallStates 30, 37
DwCallTreatmentListOffset 30
DwCallTreatmentListSize 30
DwChargingInfoOffset 37
DwChargingInfoSize 37
DwCommentOffset 37
DwCommentSize 37
DwCompletionMsgTextEntrySize 30
DwCompletionMsgTextOffset 30
DwCompletionMsgTextSize 30
DwConnectedIDAddressType 37
DwConnectedIDFlags 30
DwConnectedModes 30
DwCountryCode 37
DwDevCapFlags 40
DwDeviceClassesOffset 30
DwDeviceClassesSize 30, 40
DwDevSpecificOffset 30, 39, 40
DwDevSpecificSize 30, 36, 39, 40
DwDialToneModes 30
DwDigitMode 17, 25
DwDigitModes 25
DwDisconnectModes 30
DwDivertSettings 59, 62
DwForwardModes 30
DwForwardNumEntries 36
DwForwardOffset 36
DwForwardSize 36
DwGatherDigitsMaxTimeout 40
DwGatherDigitsMinTimeout 40
DwGenerateToneMaxNumFreq 40
DwGenerateToneModes 40
DwGroupCount 62
DwHighLevelCompOffset 37
DwHighLevelCompSize 37
DwLineDeviceID 30
DwLineFeatures 40
DwLineNameSize 40
dwLineStates 28, 40
DwLowLevelCompOffset 37
DwLowLevelCompSize 37

DwMaxCallCompletions 30
DwMaxCallDataSize 30
DwMaxForwardEntries 30
DwMaxFwdNumRings 30
DwMaxNoAnswerTimeout 30
DwMaxNumActiveCalls 30, 40
DwMaxNumConference 30
DwMaxNumOnHoldCalls 30
DwMaxNumOnHoldPendingCalls 30
DwMaxNumTransConf 30
DwMaxRate 40
DwMaxSpecificEntries 30
DwMedCtlCallStateMaxListSize 40
DwMedCtlDigitMaxListSize 40
DwMedCtlMediaMaxListSize 40
DwMedCtlToneMaxListSize 40
DwMediaMode 37
DwMediaModes 26, 40
DwMinFwdNumRings 30
DwMonitorMediaModes 37
DwMonitorToneMaxNumEntries 40
DwMonitorToneMaxNumFreq 40
DwNumActiveCalls 36
DwNumAddresses 18, 40
DwNumCallTreatments 30
DwNumCompletionMessages 30
DwNumInUse 36
DwNumMessages 62
DwNumOnHoldCalls 36
DwNumOnHoldPendCalls 36
DwNumRingsNoAnswer 36
DwNumTerminals 40
DwNumTones 17
DwOfferingModes 30
DWORD 18, 61
DWORD dwAddressID 14, 18, 20, 29
DWORD dwAddressMode 18
DWORD dwAddressStates 28
DWORD dwAPIHighVersion 25
DWORD dwAPILowVersion 25
DWORD dwAPIVersion 18, 19, 26
DWORD dwAppSpecific 27
DWORD dwCallbackInstance 26
DWORD dwCallPrivilege 27
DWORD dwCountryCode 12, 17, 24, 26
DWORD dwDeviceID 13, 18, 19, 25, 26
DWORD dwDigitMode 17
DWORD dwDigitModes 25
DWORD dwDivertSettings 59, 62
DWORD dwDuration 17
DWORD dwExtVersion 18, 19, 26
DWORD dwGroupCount 62
DWORD dwLineStates 28
DWORD dwMediaMode 19, 24, 27
DWORD dwMediaModes 26
DWORD dwNumEntries 25
DWORD dwNumMessages 62
DWORD dwNumTones 17
DWORD dwParkMode 26
DWORD dwPriority 27
DWORD dwPrivileges 26
DWORD dwRequestMode 19, 27
DWORD dwSelect 20
DWORD dwSize 12, 14, 17, 18, 48
DWORD dwToneMode 17
DWORD dwTransferMode 12
DwParkModes 30
DwPermanentLineID 40
DwPredictiveAutoTransferStates 30
DwProviderInfoSize 40
DwRate 37
DwReceivingFlowspecOffset 37
DwReceivingFlowspecSize 37
DwRedirectingIDAddressType 37
DwRedirectingIDFlags 30
DwRedirectionIDAddressType 37
DwRedirectionIDFlags 30
DwRemoveFromConfCaps 30
DwRemoveFromConfState 30
DwRingModes 40
DwSelect 20
DwSendingFlowspecOffset 37
DwSendingFlowspecSize 37
DwSettableDevStatus 40
DwSpecialInfo 30
DwStringFormat 40
DwSwitchInfoSize 40
DwTerminalCapsOffset 40
DwTerminalCapsSize 40
DwTerminalModesOffset 36, 37
DwTerminalModesSize 36, 37
DwTerminalTextEntrySize 40
DwTerminalTextOffset 40
DwTerminalTextSize 40
DwToneMode 17
  Wert 17
DwTransferModes 30
DwTrunk 37
DwUserUserInfoOffset 37
DwUserUserInfoSize 37
DwUUIAcceptSize 40
DwUUIAnswerSize 40
DwUUICallInfoSize 40
DwUUIDropSize 40
DwUUIMakeCallSize 40
DwUUISendUserUserInfoSize 40

E

eConsole 26
Eg 21
Eigenschaften 5
  IP Office 5
Eingabe 7
  IP 7
Einzelbenutzer 7
  Auswählen 7
Einzelbenutzermodus 7
Ende 24
  TAPI 24
Entparken 26
EnumerateAddresses 45
Er/Sie 60
ERFOLGREICH 59
Erfordernisse 9
  TSP 9
Erkannt 25
  LINE_MONITORDIGITS 25
Erkennung 59
  MSP 59

Ermöglicht 9, 59
TAPI 59
Telephony Service Provider (Telefonie-Dienstanbieter) 9
Extnnum 16

F

FEHLER 59
Festlegen 21, 26, 27, 28
anwendungsspezifisch 27
LINEADDRESSSTATE_ALL 28
LINEDEVSTATE_ALL 28
LINEMEDIAMODE_INTERACTIVEVOICE 26
Nicht stören 21
Festlegen von WML 15
Finish 49
FINISH_MODE finishMode 49
Fire_DivertSettingsChanged 61
ForwardAll 61
ForwardAllOff 15
ForwardAllOn 15
ForwardBusy 61
ForwardBusyOff 15
ForwardBusyOn 15
ForwardNoAnswer 61
ForwardNoAnswerOff 15
ForwardNoAnswerOn 15
Funktionscode 14, 15, 16

G

Gerätespezifisch 43
Gerätespezifische Schnittstellen 59
Verwendung 59
Get 20, 46
BSTR 46
ID 20
Get_Address 48
Get_AddressName 46
Get_Call 52, 53
Get_CallInfoString 48
Get_CallState 48, 55
Get_Cause 52, 57
Get_DialableAddress 46
Get_MediaTypes 47
Get_ServiceProviderName 46
Get_State 52, 55
GetDivertAllDestination 61
GetDivertSettings 59, 61
GpAddress->QueryInterface 59
Groupextn 60
Groupnum 16
Gruppe aktivieren 16

H

H323 46
H323.tsp 46
H323.tsp 46
H323 46
HCALL hActiveCall 28
HCALL hCall 12, 13, 14, 17, 19, 20, 24, 25, 26, 27, 28, 29
HCALL hConfCall 12
HCALL hConsultCall 12
HCALL hHeldCall 28
Heranholen beliebig (Funktionscode) 60
Ausführung 60
Heranholen Nst (Funktionscode) 60
Ausführung 60
Heranholen Sammelanschluss beliebig (Funktionscode) 60
Ausführung 60
HINSTANCE hInstance 24
HLINE hLine 14, 18, 20, 21, 24, 28, 29
HLINEAPP hLineApp 18, 19, 25, 26, 28
Hold 49
HRESULT 45, 46, 47, 48, 49, 52, 53
HRESULT BlindTransfer 49
HRESULT Conference 49
HRESULT Finish 49
HRESULT RegisterCallNotifications 45
HRESULT RemoveFromConference 49
HRESULT Transfer 49
HWND hwndOwner 13

I

ID 20, 30
Get 20
Umleitung 30
Ie 16
IEnumAddress 45, 47
Ihr IP Office 8
Konfigurieren 8
IID_ITDIVERT 59
Info 59
MSP 59
Initialisierung von TAPI 24, 45
Angerufen 24, 45
Installieren 7
CTI TAPI Linkpro-Lizenz 7
TAPILink 7
Int len 14, 15, 16
Interfaces.h (Datei) 59
DevSpice 59
IP 7
Eingabe 7
IP Office 5, 7, 8, 9, 15, 18, 25, 26, 29, 53, 59, 63
ITAddressEvent 9
Verhalten 5
Verwendet 7
IP Office CTI Link 5
IP Office CTI Link Pro 5
IP Office CTI Link-Installationshandbuch 5
Bezug auf 5
IP Office Media Service Provider 59
IP Office privat 59
IP Office TAPI 30, 40
beziehen 30, 40
IP Office TAPI-Serviceanbieter 5, 7
Verwendet 5
IP Office TAPI-Treiber 5
Versionen 5
IP Office-Benutzer-CD 7
IP Office-MSP 63
IP_OFFICE_DIVERT_SETTINGS 61
IP_OFFICE_DIVERT_SETTINGS enum 61, 62
IP401 7
IP403 7
IP406 7
IP412 7

IP-Adresse der Vermittlung 7
IPOFF_DESTINATION 61
IPOFF_DND 61
IPOFF_FWDALL 61
IPOFF_FWDBUSY 61
IPOFF_NOANSWER 61
IPOfficePrivateEvent 59
IPOfficePrivateEvents 61, 62
ISDN 26
IsLoggedIn 60
ITACDAgent 59, 60
ITAddress 45, 46, 47, 48, 59
ITAddressEvent 9
  IP Office 9
ITBasicCallControl 46, 49, 57
ITCallInfo 48, 52, 53, 55, 56
  ausgeben 53
ITCallInfoChangeEvent 53
ITCallNotificationEvent 53
ITCallStateEvent 52, 55, 57
ITDivert 59, 61
ITGroup 59, 60
ITMediaSupport 47
ITPlay 59, 62
ITTAPI 45

K

Klingelton/aktiv 29
Kommunikationsverlust 5
Konferenz 30, 39, 49
Konfigurieren 7, 8
  Ihr IP Office 8
  TAPI-Treiber 7
Konstante 14
Kontrollkästchen 7, 8

L

LCallbackInstance 45
LED 15
Leitungen/Adressen 9
Leitungsname 40
  Anzeige 40
LFilterMask 45
LINE FORWARDMODE_BUSY 30
LINE FORWARDMODE_BUSYEXTERNAL 30
LINE FORWARDMODE_BUSYINTERNAL 30
LINE FORWARDMODE_BUSYNA 30
LINE FORWARDMODE_BUSYNAEXTERNAL 30
LINE FORWARDMODE_BUSYNAINTERNAL 30
LINE FORWARDMODE_BUSYNASPECIFIC 30
LINE FORWARDMODE_BUSYSPECIFIC 30
LINE FORWARDMODE_NOANSW 30
LINE FORWARDMODE_NOANSWEXTERNAL 30
LINE FORWARDMODE_NOANSWINTERNAL 30
LINE FORWARDMODE_NOANSWSPECIFIC 30
LINE FORWARDMODE_UNCOND 30
LINE FORWARDMODE_UNCONDEXTERNAL 30
LINE FORWARDMODE_UNCONDSPECIFIC 30
LINE_ADDRESSSTATE 43
LINE_APPNEWCALL 43
LINE_CALLINFO 43
LINE_CALLSTATE 43
LINE_DEVSPECIFIC 43
LINE_GENERATE 17
LINE_LINEDEVSTATE 43
LINE_MONITORDIGITS 25
  Erkannt 25
LINE_MONITORTONE 25
LINEADDRCAPFLAGS_CONFERENCEHELD 30
LINEADDRCAPFLAGS_CONFERENCEMAKE 30
LINEADDRCAPFLAGS_DIALED 30
LINEADDRCAPFLAGS_FWDNUMRINGS 30
LINEADDRCAPFLAGS_FWDSTATUSVALID 30
LINEADDRCAPFLAGS_TRANSFERHELD 30
LINEADDRCAPFLAGS_TRANSFERMAKE 30
LINEADDRESSCAPS 18, 30, 39, 43
LINEADDRESSMODE_ADDRESSID 40
LINEADDRESSSHARING_PRIVATE 30
LINEADDRESSSTATE_ALL 28
  einrichten 28
LINEADDRESSSTATUS 36
LINEADDRFEATURE_FORWARD 30
LINEADDRFEATURE_FORWARDDND 30
LINEADDRFEATURE_FORWARDFWD 30
LINEADDRFEATURE_MAKECALL 30, 36
LINEADDRFEATURE_SETUPCONF 30, 36
LINEADDRFEATURE_UNPARK 30, 36
LineAddToConference 12
LineAnswer 12
LINEANSWERMODE_NONE 40
LINEBEARERMODE_SPEECH 40
LINEBEARERMODE_VOICE 37, 38, 40
LineBlindTransfer 12
LINEBUSYMODE_UNAVAIL 30
LINECALLBACK lpfnCallback 24
LINECALLFEATURE_ADDTOCONF 30
LINECALLFEATURE_ANSWER 30
LINECALLFEATURE_BLINDTRANSFER 30
LINECALLFEATURE_COMPLETETRANSF 30
LINECALLFEATURE_DIAL 30
LINECALLFEATURE_DROP 30
LINECALLFEATURE_GENERATEDIGITS 30
LINECALLFEATURE_HOLD 30
LINECALLFEATURE_PARK 30
LINECALLFEATURE_REDIRECT 30
LINECALLFEATURE_REMOVEFROMCONF 30
LINECALLFEATURE_SETCALLDATA 30
LINECALLFEATURE_SETUPTRANSFER 30
LINECALLFEATURE_SWAPHOLD 30
LINECALLFEATURE_UNHOLD 30
LINECALLINFO 37, 43
LINECALLINFOSTATE_CALLDATA 30
LINECALLINFOSTATE_CALLEDID 30
LINECALLINFOSTATE_CALLERID 30
LINECALLINFOSTATE_CALLID 30
LINECALLINFOSTATE_DISPLAY 30

LINECALLINFOSTATE_MONITORMODES 30
LINECALLINFOSTATE_NUMMONITORS 30
LINECALLINFOSTATE_NUMOWNEDECR 30
LINECALLINFOSTATE_NUMOWNERINCR 30
LINECALLINFOSTATE_REDIRECTINGID 30
LINECALLINFOSTATE_REDIRECTIONID 30
LINECALLINFOSTATE_RELATEDCALLID 30
LINECALLPARAMFLAGS_IDLE 37
LINECALLPARAMS 38
LINECALLPARTYID_ADDRESS 30
LINECALLPARTYID_BLOCKED 30
LINECALLPARTYID_NAME 30
LINECALLPARTYID_OUTOFAREA 30
LINECALLPARTYID_UNAVAIL 30
LINECALLPARTYID_UNKNOWN 30
LINECALLSELECT_LINE 20
LINECALLSTATE_ACCEPTED 30, 39
LINECALLSTATE_BUSY 30, 37, 39
LINECALLSTATE_CONFERENCED 30, 37, 39
LINECALLSTATE_CONNECTED 30, 37, 39
LINECALLSTATE_DIALING 30, 37, 39
LINECALLSTATE_DIALTONE 30, 37, 39
LINECALLSTATE_DISCONNECTED 30, 37, 39
LINECALLSTATE_IDLE 30, 37, 39
LINECALLSTATE_OFFERING 30, 37, 39
LINECALLSTATE_ONHOLD 30, 37, 39
LINECALLSTATE_ONHOLDPENDCONF 30, 37, 39
LINECALLSTATE_ONHOLDPENDTRANSFER 30, 37, 39
LINECALLSTATE_PROCEEDING 30, 37, 39
LINECALLSTATE_RINGBACK 30, 37, 39
LINECALLSTATE_UNKNOWN 30, 37, 39
LINECALLSTATUS 19, 39
  Abrufen 19
LineCompleteTransfer 12
LineConfigDialog 13
LineDeallocateCall 13
LINEDEVCAPFLAGS_CLOSEDROP 40
LINEDEVCAPFLAGS_DIALBILLING 40
LINEDEVCAPFLAGS_DIALDUALTONE 40
LINEDEVCAPFLAGS_DIALQUIET 40
LINEDEVCAPS 18, 19, 40
LineDevSpecific 6, 7, 14, 15
  Anrufen 14
  Puffer 15
LINEDEVSTATE_ALL 28
  einrichten 28
LINEDEVSTATE_CLOSE 40
LINEDEVSTATE_CONNECTED 40, 43
LINEDEVSTATE_DEVSPECIFIC 43
LINEDEVSTATE_DISCONNECTED 40, 43
LINEDEVSTATE_INSERVICE 9, 40, 43
LINEDEVSTATE_OPEN 40
LINEDEVSTATE_OUTOFSERVICE 9, 40, 43
LINEDEVSTATE_REINIT 40
LINEDEVSTATE_REMOVED 40
LINEDEVSTATE_RINGING 40, 43
LINEDEVSTATE_TRANSLATECHNGE 40
LineDial 17
LINEDIALTONEMODE_UNAVAIL 30
LINEDIGITMODE_DTMF 17
LINEDIGITMODE_DTMF und/oder LINEDIGITMODE_DTMFEND 25
LINEDISCONNECTMODE_BADADDRESS 30
LINEDISCONNECTMODE_BLOCKED 30
LINEDISCONNECTMODE_BUSY 30
LINEDISCONNECTMODE_CONGESTION 30
LINEDISCONNECTMODE_DONOTDISTURB 30
LINEDISCONNECTMODE_FORWARDED 30
LINEDISCONNECTMODE_INCOMPATIBLE 30
LINEDISCONNECTMODE_NOANSWER 30
LINEDISCONNECTMODE_NODIALTONE 30
LINEDISCONNECTMODE_NORMAL 30
LINEDISCONNECTMODE_PICKUP 30
LINEDISCONNECTMODE_QOSUNAVAIL 30
LINEDISCONNECTMODE_REJECT 30
LINEDISCONNECTMODE_UNAVAIL 30
LINEDISCONNECTMODE_UNREACHABLE 30
LineDrop 17
LINEERR_RESOURCEUNAVAIL 26
LINEFEATURE_MAKECALL 40
LineGenerateDigits 17
LineGenerateTone 17
LineGetAddressCaps 18, 30
LineGetAddressID 18
LineGetAddressStatus 18, 36
LineGetAppPriority 19
LineGetCallInfo 19, 37
LineGetCallStatus 19, 39
LineGetDevCaps 19, 40
LineGetID 20
LineGetLineDevStatus 6, 21, 43
LineHandoff 24
LineHold 24
LineInitializeEx 24, 25
LineMakeCall 24, 38
LINEMEDIAMODE_INTERACTIVEVOICE 26, 37, 40
  einrichten 26
LINEMEDIAMODE_UNKNOWN 26
  Analog 26
LineMonitorDigits 25
LineMonitorTone 25
LineNegotiateAPIVersion 25
LineOpen 26
LinePark 26, 29
LINEPARKMODE_DIRECTED 26, 30
LineRedirect 26
LINEREMOVEFROMCONF_ANY 30
LineRemoveFromConference 27
LineSetAppPriority 27
LineSetAppSpecific 27
LineSetCallPrivilege 27
LineSetStatusMessages 28
LineSetupTransfer 28, 38
LineShutdown 28
LINESPECIALINFO_UNAVAIL 30
LineSwapHold 28
LINETONEMODE_BEEP 17, 40
LINETRANSFERMODE_CONFERENCE 30
LINETRANSFERMODE_TRANSFER 30
LineUnhold 29

LineUnpark 29
Lite 5, 7
LMediaTypes 45
Login erzwingen 21
LONG 12, 13, 14, 17, 18, 19, 20, 21, 24, 25, 26, 27, 28, 29
Long 1AddressType 46
Long 1MediaTypes 46
LPCSTR lpsAddress 18
LPCSTR lpsUserUserInfo 12, 17
LPCSTR lpszAppFilename 19, 27
LPCSTR lpszDestAddress 12, 17, 24, 26, 29
LPCSTR lpszDeviceClass 13, 20
LPCSTR lpszDigits 17
LPCSTR lpszDirAddress 26
LPCSTR lpszExtensionName 27
LPCSTR lpszFileName 24
LPCSTR lpszFriendlyAppName 24
LpdwAddressID 18
LpdwAPIVersion 24
LPDWORD lpdwAddressID 18
LPDWORD lpdwAPIVersion 24, 25
LPDWORD lpdwNumDevs 24
LPDWORD lpdwPriority 19
LPHCALL lphCall 24, 29
LPHCALL lphConfCall 12
LPHCALL lphConsultCall 28
LPHLINE lphLine 26
LPHLINEAPP lphLineApp 24
LPLINEADDRESSCAPS lpAddressCaps 18
LPLINEADDRESSSTATUS lpAddressStatus 18
LPLINECALLINFO lpCallInfo 19
LPLINECALLPARAMS const lpCallParams 24, 26, 28
LPLINECALLSTATUS lpCallStatus 19
LPLINEDEVCAPS lpLineDevCaps 19
LPLINEDEVSTATUS lpLineDevStatus 21
LPLINEEXTENSIONID lpExtensionID 19, 25, 27
LPLINEGENERATETONE const lpTones 17
LPLINEINITIALIZEEXPARAMS lpLineInitializeExParams 24
LPLINEMONITORTONE const lpToneList 25
LpszDestAddress 29
lpszDeviceClass (Parameter) 13
LpToneList 25
LPVARSTRING lpDeviceID 20
LPVARSTRING lpExtensionName 19
LPVARSTRING lpNonDirAddress 26
LPVOID lpParams 14

M

M+1 21
M+2.p 21
Maibox-Nachrichten 15
Manager 7, 8, 16, 60, 62
Manager-Anwendung 8
Media Service Provider 5
Media-Streaming-Funktionen 63
  Verwendung 63
Microsoft Developer Network 5
  Zugreifen 5
Microsoft TAPI 30, 36, 37, 38, 39, 40
Modem 13
MSDN 5, 59, 63
MSP 59, 62, 63
  Erkennung 59
  Info 59
  verwenden 59
MSP Call 62

N

N+1 21
N+10 21
N+11 21
N+12 21
N+13 21
N+14 21
N+15 21
N+16 21
N+17 21
N+18 21
N+19 21
N+2 21
N+20 21
N+21 21
N+22 21
N+23 21
N+24 21
N+25 21
N+26 21
N+27 21
N+28.m 21
N+3 21
N+4 21
N+5 21
N+6 21
N+7 21
N+8 21
N+9 21
Nachschlagen 5
Name 40
  TSP 40
Neue TAPI 6
Next 47
Nicht stören 15, 21, 61, 62
  einrichten 21
Nicht stören aus 15
Nicht stören ein 15
NOANSWER 57
Nst 21
NULL 25, 59
Nullabschlusszeichen 21
Num 15

O

OnDivertSettingsChanged 62
OnGroupChanged 62
OnUserLogin 62
OnUserLogout 62
OnVoiceMail 62

P

P+1 21
P+2.q 21
PAddress 45
ParkDirect 49
PC 7
PCallInfoBuffer 48
PCallState 48, 52

PCEC 52
PceltFetched 47
PDestination 61
PDialableAddress 46
PDivert 59
PdwDivertSets 61
PermanentLineGuide 40
Phone Manager 26
PickupAny 60
PickupExtn 60
PickupGroup 60
PickupMembers 60
Plattform-SDK 5
PlMediaTypes 47
PlRegister 45
PpAddress 48
PpCall 46, 53
PpCallInfo 52
PpCallInfoString 48
PpElements 47
PpEnumAddress 45
PpName 46
PrAnr 49
Pro 5
ProtocolGuide 40
Puffer 14, 15
  DevSpecific 14
  lineDevSpecific 15
Put_EventFilter 45

Q

Q+1 21
Q+2.r 21
QueryInterface 59

R

R+1 21
R+2.s 21
RegisterCallNotifications 45
RemoveFromConference 49
Returns S_TRUE 60
Rückruf 21
Rufweiterleitung 15
RUHEND 48

S

S_FALSE 60
S+1 21
S+2.t 21
SDK 5
SDK-CD 59
SetCallInfoBuffer 48
SetDivertAllDestination 61
SetDoNotDisturb 61
SetForwardAll 61
SetForwardBusy 61
SetForwardNoAnswer 61
Siehe 62
  DevSpice 62
Siehe dwCallStates 43
Siehe LINEADDRESSCAPS 18
Siehe LINEADDRESSSTATUS 18
Siehe LINECALLINFO 19
Siehe LINECALLSTATUS 19
Siehe LINEDEVCAPS 19
Siehe Manager 60
Small Office Edition 7
Softwareentwicklungs-Kit 5
Sprintf 15, 16
StartPlay 62
StartRecord 62
StopPlay 62
StopRecord 62
STRINGFORMAT_ASCII 40
SwapHold 49
Systemsteuerung 13

T

T+1 21
T+2.u 21
T1 26
TAPI 5, 7, 8, 9, 14, 18, 19, 24, 25, 28, 45, 59, 63
  angeben 7
  Anrufe 9
  end 24
  Zulässig 59
TAPI 1.x 5
TAPI 3.0 45, 48, 59
TAPI 3.1 48, 62
TAPI auf Benutzer-CD 1.4 von IP Office 6
TAPI lineDevSpecific 14
TAPI Version 2.2 40
TAPI Version 3.0 37, 40
TAPI WAV 7
TAPI/Leitung 40
TAPI-Angaben 39
TAPI-Anwendung 7, 9
  Anzeige 9
TAPI-Dienstanbieter 13
TAPI-Ereignisse 43
TAPILink 7
  Installieren 7
TAPI-Treiber 7
  Konfigurieren 7
TCP 9
Telefon 13
Telefonie 8, 13
Telefonie-Dienstanbieter 5, 9, 46
  Zulässig 9
Telefonnebenstelle 21
Terminal 47
Timeout 21
Tonwahl 17, 25
Transfer 49
TSP 5, 9, 14, 40, 43, 46
  Erforderlich 9
  Name 40
TSPI 14
TSPs 5
TStateEntryTime 39
Typedef enum 61

U

U+1 21
U+2 21
U+3 21
U+4 21
U+5 21
U+6 21
U+7 21

Überblick 5, 9
Überprüfen 7
  ACD-Warteschlangen (Kontrollkästchen) 7
ULONG 47
Umleiten 21
  Voicemail 21
Umleiten wenn besetzt 62
Umleitung 30
  ID 30
Umleitungsziel 15
und/oder 7
Unimdm.tsp 46
  Unimodem 46
Unimodem 46
  Unimdm.tsp 46
Unpark 49
UserUserInfo 12, 17

V

VARIANT_BOOL bOn 61
VARIANT_BOOL fHold 49
VARIANT_BOOL fMonitor 45
VARIANT_BOOL fOwner 45
VARIANT_BOOL fSync 49
Verrechnungscode einrichten 60
Versionen 5
  IP Office TAPI-Treiber 5
Verwenden des MSP 59
Verwendet 5, 7, 59
  IP Office 7
  IP Office TAPI-Serviceanbieter 5
  MSP 59
  Wave 7
Verwendung 59, 63
  Gerätespezifische Schnittstellen 59
  Media-Streaming-Funktionen 63
VoiceMail 7, 15, 21, 62
  Nummer 21, 62
  Umleit 21
Voicemail/E-Mil 21
VoiceMail-Rückruf 21

W

WAV 7, 8, 17, 25, 59
WAV-Benutzer 7, 26
WAV-Benutzer (Kontrollkästchen) 7
Wave 7, 26, 59
  verwenden 7
Wave-Benutzer 26
Wave-Benutzerlizenz 7
Wave-Lizenzen 7
Wave-Treiber 7
Weitergeben 48
  CIB_CALLDATABUFFER 48
Wert 17
  dwToneMode 17
Wiederherstellung 5
WINAPI 12, 13, 14, 17, 18, 19, 20, 21, 24, 25, 26, 27, 28, 29
Windows 5, 40
Windows Telephony Programming 5
Windows XP 48

Z

Zahl 21, 62
  Voicemail 21, 62
Zugriff 5
  Microsoft Developer Network 5

Die Leistungsdaten und Angaben in diesem Dokument sind typisch und müssen ausdrücklich schriftlich von Avaya bestätigt werden, bevor sie auf eine Bestellung oder einen Auftrag angewendet werden dürfen. Änderungen und Ergänzungen an den ausführlichen Spezifikationen vorbehalten. Die Veröffentlichung der Informationen in diesem Dokument entbindet den Leser nicht von den Patentrechten oder anderen Schutzrechten der Firma Avaya oder anderen Firmen.

Das geistige Eigentum an diesem Produkt (einschließlich Marken), das für Lucent Technologies eingetragen wurde, ist an Avaya weitergegeben oder lizenziert.

Alle durch ® oder ™ gekennzeichneten Marken sind Marken bzw. eingetragene Marken von Avaya Inc. Alle anderen Marken sind das Eigentum ihrer jeweiligen Inhaber.

Dieses Dokument enthält firmeneigene Informationen von Avaya, die nur vertragsgemäß weitergegeben und verwendet werden dürfen.

Anmerkungen oder Vorschläge bezüglich dieses Dokuments können an wgctechpubs@avaya.com gesendet werden.

© 2004 Avaya Inc. Alle Rechte vorbehalten.

Avaya
Sterling Court
15 - 21 Mundells
Welwyn Garden City
Hertfordshire
AL7 1LZ
England

Tel: +44 (0) 1707 392200
Fax: +44 (0) 1707 376933

Web: http://www.avaya.com